

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Yasunori KANEDA et al.

Serial No: Not assigned

Filed: September 25, 2003

For: Method for Allocating Storage Regions and
Performance Guarantee Method Based on Hints,
Storage Device and Management Program

Art Unit: Not assigned

Examiner: Not assigned

TRANSMITTAL OF PRIORITY DOCUMENT

Mail Stop PATENT APPLICATION
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Enclosed herewith is a certified copy of Japanese patent application No. 2003-189157 which was filed July 1, 2003, from which priority is claimed under 35 U.S.C. § 119 and Rule 55.

Acknowledgment of the priority document(s) is respectfully requested to ensure that the subject information appears on the printed patent.

Respectfully submitted,

HOGAN & HARTSON L.L.P.

Date: September 25, 2003

By: 

Anthony J. Orler
Registration No. 41,232
Attorney for Applicant(s)

500 South Grand Avenue, Suite 1900
Los Angeles, California 90071
Telephone: 213-337-6700
Facsimile: 213-337-6701

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 7 月 1 日
Date of Application:

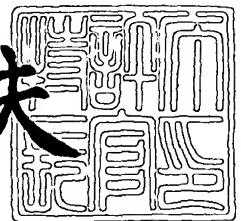
出 願 番 号 特 願 2 0 0 3 - 1 8 9 1 5 7
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 1 8 9 1 5 7]

出 願 人 株 式 会 社 日 立 製 作 所
Applicant(s):

2 0 0 3 年 7 月 3 1 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 6 1 1 6 7

【書類名】 特許願

【整理番号】 K03005371A

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/06

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 兼田 泰典

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 藤田 高広

【発明者】

【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所システム開発研究所内

【氏名】 中川 弘隆

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】	要約書 1
【プルーフの要否】	要

【書類名】 明細書

【発明の名称】 ヒントに基づく記憶領域の割当と性能保証方法、記憶装置及び管理プログラム

【特許請求の範囲】

【請求項 1】 計算機によりアクセスされるデータを保持可能な記憶領域と、前記計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジュールとを有する記憶装置であって、

管理計算機から、前記計算機に前記記憶領域を割当てる要求を受信する管理インタフェースと、

前記記憶領域割当要求に従って、前記計算機に未割当の記憶領域を計算機によりアクセス可能なように割当てる記憶領域割当モジュールと、

前記記憶領域割当要求に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当てる性能割当モジュールを有する記憶装置。

【請求項 2】 請求項 1 に記載の記憶装置であって、

前記対象モジュールには前記計算機と接続するためのインタフェースが含まれ

、
前記性能割当モジュールは、前記インタフェースにおける帯域性能を制御する帯域制御モジュールであることを特長とする記憶装置。

【請求項 3】 請求項 1 に記載の記憶装置であって、

前記計算機との間のアクセスされるデータを一時保持するキャッシュと、

前記性能割当モジュールは、前記記憶領域割当要求に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する前記キャッシュの割当量を制御するキャッシュ割当制御モジュールであることを特長とする記憶装置。

【請求項 4】 請求項 1 に記載の記憶装置であって、

前記記憶領域管理モジュールは、管理計算機からの前記割当済の記憶領域に対する割当開放要求に従って、前記割当済の記憶領域を未割当の記憶領域に設定し

、
前記性能割当モジュールは、前記割当開放要求に従って、前記対象モジュール

に対して設定した性能の保証を解除することを特長とする記憶装置。

【請求項 5】 計算機によりアクセスされるデータを保持可能な記憶領域と、前記計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジュールとを有する記憶装置であって、

管理計算機から、前記計算機に前記記憶領域を割当て要求を受信する管理インタフェースと、

前記記憶領域割当て要求に従って、前記未割当の記憶領域を計算機によりアクセス可能なように割当てする記憶領域割当モジュールと、

前記記憶領域割当て要求に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する性能を割当てるように外部情報処理装置へ指示する性能設定モジュールを有する記憶装置。

【請求項 6】 請求項 1 に記載の記憶装置であって、

前記性能割当モジュールは、前記記憶領域割当て要求と前記対象モジュールの実行性能値とに従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する性能を割当てるように前記対象モジュールを制御することを特長とする記憶装置。

【請求項 7】 請求項 1 に記載の記憶装置であって、

前記記憶領域割当て要求には、割当て要求の対象である前記記憶領域に対するアクセス帯域幅、アクセスランダム性又は、アクセス応答時間に関する方針が含まれおり、

前記性能割当モジュールは、前記割当て要求の対象である前記記憶領域に対する方針に基づき変換される性能パラメータに基づいて、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当ることを特長とする記憶装置。

【請求項 8】 請求項 1 に記載の記憶装置であって、

複数の前記記憶領域を管理する複数のディスクドライブをさらに有し、

前記記憶領域割当て要求には、割当て要求の対象である前記記憶領域に対するアクセスランダム性に関する方針が含まれおり、

前記記憶領域割当モジュールは、前記記憶領域割当て要求に含まれるランダムア

アクセス性に関する方針と前記ディスクドライブに既に割当てられている記憶領域のランダムアクセス性能情報とに従って、前記未割当の記憶領域を割当てる前記ディスクドライブを特定することを特長とする記憶装置。

【請求項 9】 請求項 1 に記載に記憶装置であって、

前記記憶領域割当要求には、割当て要求の対象である前記記憶領域に対するアクセスランダム性とアクセス帯域幅に関する方針が含まれおり、

前記記憶領域割当モジュールは、前記記憶領域割当要求に含まれるランダムアクセス性とアクセス帯域幅に関する方針と前記ディスクドライブに既に割当てられている記憶領域のランダムアクセス性能情報とアクセス帯域幅とに従って、前記未割当の記憶領域を割当てる前記ディスクドライブを特定することを特長とする記憶装置。

【請求項 10】 請求項 1 に記載の記憶装置であって、

複数の前記記憶領域を管理する複数のディスクドライブをさらに有し、

前記記憶領域割当要求には、割当て要求の対象である前記記憶領域に対するアクセスランダム性に関する方針が含まれおり、

前記記憶領域割当モジュールは、前記記憶領域割当要求に含まれるランダムアクセス性に関する方針と前記ディスクドライブに既に割当てられている記憶領域のランダムアクセス性能とに従って、前記未割当の記憶領域を割当てる前記計算機と接続するためのインタフェースを特定することを特長とする記憶装置。

【請求項 11】 請求項 1 に記載に記憶装置であって、

前記記憶領域割当要求には、割当て要求の対象である前記記憶領域に対するアクセスランダム性とアクセス帯域幅に関する方針が含まれおり、

前記記憶領域割当モジュールは、前記記憶領域割当要求に含まれるランダムアクセス性とアクセス帯域幅に関する方針と前記ディスクドライブに既に割当てられている記憶領域のランダムアクセス性能とアクセス帯域幅とに従って、前記未割当の記憶領域を割当てる前記計算機と接続するためのインタフェースを特定することを特長とする記憶装置。

【請求項 12】 計算機によりアクセスされるデータを保持可能な記憶領域と、前記計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジ

ルールとを有する記憶装置における性能割当方法であって、

管理計算機から、前記計算機に前記記憶領域を割当てて要求を受信するステップと、

前記記憶領域割当要求に従って、前記計算機に未割当の記憶領域を計算機によりアクセス可能なように割当ててステップと、

前記記憶領域割当要求に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当ててステップとを有する性能保証方法。

【請求項 13】 計算機と、前記計算機によりアクセスされるデータを保持可能な記憶領域と前記計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジュールとを有する記憶装置とに接続された管理計算機における管理方法であって、

前記計算機に前記記憶領域を割当てて要求を受付けるステップと、

前記記憶領域割当要求に従って、前記計算機に未割当の記憶領域を計算機によりアクセス可能なように割当ててように前記記憶装置に指示するステップと、

前記記憶領域割当要求に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当ててように性能割当モジュールに指示するステップとを有する管理方法。

【請求項 14】 計算機と、前記計算機によりアクセスされるデータを保持可能な記憶領域と前記計算機との間でのデータのアクセスを制御するアクセス処理モジュールとを有する記憶装置とに接続された管理計算機に実行させるための管理プログラムであって、

前記計算機に前記記憶領域を割当てて要求を受付ける手順と、

前記記憶領域割当要求に従って、前記計算機に未割当の記憶領域を計算機によりアクセス可能なように割当ててように前記記憶装置に指示する手順と、

前記記憶領域割当要求に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当ててよう前記記憶装置に指示する手順とを前記管理計算機に実行させるプログラム。

【請求項 15】 計算機と、前記計算機によりアクセスされるデータを保持可能

な記憶領域と前記計算機との間でのデータのアクセスを制御するアクセス処理モジュールとを有する記憶装置とに接続された管理計算機に実行させるための管理プログラムであって、

前記計算機に前記記憶領域を割当てて要求を受付ける手順と、

前記記憶装置から、記憶装置内部の複数の対象モジュールの実行性能値を受信する手順と、

前記記憶領域割当要求に従って、前記計算機に未割当の記憶領域を計算機によりアクセス可能なように割当てて前記記憶装置に指示する手順と、

前記記憶領域割当要求と前記対象モジュールの実行性能値とに従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する性能を割当てるように前記対象モジュールを特定し、かつ、特定した対象モジュールの性能を割当てて前記記憶装置に指示する手順とを前記管理計算機に実行させるプログラム。

【請求項 1 6】請求項 1 4 に記載のプログラムであって、

前記管理計算機とは別の第2の管理計算機から、前記割当てて要求を受信するように前記管理計算機に実行させるプログラム。

【請求項 1 7】計算機と記憶装置と第1の管理計算機と第2の管理計算機とがネットワークを介して接続されたネットワークシステムであって、

前記第 2 の管理計算機は、

前記第1の管理計算機から、前記計算機に前記記憶領域を割当てて要求を受付ける受付モジュールと、

前記記憶領域割当要求に従って、前記計算機に未割当の記憶領域を計算機によりアクセス可能なように割当てて前記記憶装置に指示する制御モジュールと、

前記記憶領域割当要求に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当てて前記記憶装置に指示する性能割当モジュールとを有し、

前記記憶装置は、

前記計算機によりアクセスされるデータを保持可能な記憶領域と、

前記計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジュールと、

前記第2の管理計算機から、前記記憶領域を割当てる要求と前記性能を割当て指示とを受信する管理インタフェースと、

前記記憶領域割当要求に従って、前記計算機に未割当の記憶領域を計算機によりアクセス可能なように割当てる記憶領域割当モジュールと、

前記性能を割当て指示に従って、前記計算機と前記割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当てる帯域制御モジュールとを有するネットワークシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、記憶装置における記憶領域の割当制御技術に関する。

【0002】

【従来の技術】

ディスクアレイ装置に代表される大規模な記憶装置においては、記憶装置内に計算機が情報を格納するための記憶領域を作成するためのモジュールが用意されていた。しかし、記憶領域を作成するためのモジュールは、記憶装置を製造するベンダごとにそのインタフェースや要求のフォーマットが異なっていた。このため、異なるベンダの記憶装置を導入した場合には、記憶装置の管理者は、ベンダごとに異なるインタフェースや要求のフォーマットを覚えなければならず、大規模なコンピュータシステムでは、大きな問題となっていた。そこで、DMTF（Distributed Management Task Force）は、CIM（Common Information Model）とWBEM（Web-Based Enterprise Management）を策定し、記憶装置において記憶領域を作成する際のインタフェースや要求のフォーマットを策定した。このCIMで定義されたインタフェースを用いることで、ベンダの異なる記憶装置においても、統一的な手法で記憶領域を作成することができる。このCIMで定義された記憶領域の作成インタフェースと要求のフォーマットでは、「ヒント」を用いた記憶領域の作成が定義されている。

(Device27_StorageServices.mof より抜粋)

```
// =====
// StorageSettingWithHints// =====
=====
[Experimental, Version("2.7.1"), Description (
    "This subclass of StorageSetting allows a client to specify "
    "'hint's for optimization of the volume performance. The effect "
    "of these hints is implementation dependent.") ]
class CIM_StorageSettingWithHints: CIM_StorageSetting {
    [MinValue (0), MaxValue (10), Description (
        "This hint is an indication from a client of the importance "
        "placed on data availability. Values are 0=Don't Care to "
        "10=Very Important.") ]
    uint16 DataAvailabilityHint;
    [MinValue (0), MaxValue (10), Description (
        "This hint is an indication from a client of the randomness "
        "of accesses. Values are 0=Entirely Sequential to "
        "10=Entirely Random.") ]
    uint16 AccessRandomnessHint;
    [MinValue (0), MaxValue (10), Description (
        "This hint is an indication from a client of the direction "
        "of accesses. Values are 0=Entirely Read to "
        "10=Entirely Write.") ]
    uint16 AccessDirectionHint;
    [Description (
        "This hint is an indication from a client of the optimal "
        "access sizes. Several sizes can be specified."),
        Units ("MegaBytes") ]
    uint16 AccessSizeHint[];
```

```
[MinValue (0), MaxValue (10), Description (
    "This hint is an indication from a client how important "
    "access latency is. Values are 0=Don't Care to "
    "10=Very Important.") ]
uint16 AccessLatencyHint;
[MinValue (0), MaxValue (10), Description (
    "This hint is an indication from a client of bandwidth "
    "prioritization. Values are 0=Don't Care to "
    "10=Very Important.") ]
uint16 AccessBandwidthWeight;
[MinValue (0), MaxValue (10), Description (
    "This hint is an indication of the importance the client "
    "places on the cost of storage. Values are 0=Don't Care to "
    "10=Very Important. A StorageVolume provider might choose "
    "to place data on low cost or high cost drives based on "
    "this parameter.") ]
uint16 StorageCostHint;
[MinValue (0), MaxValue (10), Description (
    "This hint is an indication of the importance placed on "
    "storage efficiency by the client. Values are 0=Don't Care "
    "to 10=Very Important. A StorageVolume provider might choose "
    "different RAID levels based on this hint.") ]
uint16 StorageEfficiencyHint;
};
```

「ヒント」を用いることで、作成する記憶領域を抽象的に指定することが出来る。「ヒント」を用いて記憶領域の作成を指示することができることにより、各ベンダの記憶装置のハードウェア構成がまったく異なっているにもかかわらず、統一的な要求のフォーマットを提供できるメリットが生まれている。

【0 0 0 3】

一方、計算機と記憶領域の間の帯域を制御するための手法として、例えば、特開 2 0 0 2 - 1 0 8 5 6 7 がある。また、I P (Internet Protocol) においては、R S V P (Resource ReSerVation Protocol (RFC2205)) が I E T F (The Internet Engineering Task Force) において策定されている。

【 0 0 0 4 】

【特許文献 1】特開 2 0 0 1 - 1 0 8 5 6 7 号

【非特許文献 1】Device27_StorageServices.mof (6 6 2 ~ 7 2 1 行目)、[online]、2 0 0 2 年 1 1 月 1 2 日、Destributed Management Task Force、[2 0 0 3 年 1 月 1 8 日検索]、インターネット<URL : http://www.dmtf.org/standards/cim_schema_v27.php>

【非特許文献 2】RFC2205.txt、[online]、1 9 9 7 年 9 月、The Internet Engineering Task Force、[2 0 0 3 年 1 月 1 8 日検索]、インターネット<URL : <http://www.ietf.org/rfc/rfc2205.txt>>

【 0 0 0 5 】

【発明が解決しようとする課題】

従来技術に基づいて帯域制御の効いた記憶領域を作成するためには、まず、DMTF の C I M で定義されたインタフェースと要求のフォーマット、ヒントを用いて記憶領域を作成する。ここで、管理者が指定するヒントには、計算機から作成する記憶領域をどのように利用したいかの設計思想 (ポリシー) がこめられている。次に、計算機と記憶領域の帯域を設定するために、記憶装置に帯域を設定する。しかし、記憶装置に対する帯域の設定に関しては、先の DMTF が定義する C I M には定義されていないため、ベンダごとに異なるインタフェースと要求のフォーマットを駆使して設定しなければならない。また、記憶領域を作成する際に指定したヒントに合致するように、帯域制御の設定パラメータを選択し、設定する必要がある。よって、管理者には、記憶領域を作成する際に指定したヒントに対する帯域設定のパラメータを選択するノウハウが必要になり、高度な知識が要求されることとなっていた。また、帯域設定のパラメータの選択を誤る危険性もあり、それを防ぐことはできない。

【 0 0 0 6 】

本発明は、上記の点を鑑みて考えられたものであり、本発明の目的は、記憶領域を割当ての際に指定する記憶領域に対する領域割当要求指針（ヒント）を用いて記憶領域を割当てるとともに、このヒントを用いて帯域制御等の性能保証をも実行することにより、管理者による割当てた記憶領域に対する計算機からのアクセス性能を保証するための設定を自動化することにある。

【0007】

【課題を解決するための手段】

本発明の一実施形態の記憶装置は、計算機によりアクセスされるデータを保持可能な記憶領域と、計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジュールとを有する。管理計算機から、計算機に前記記憶領域を割当てた要求を受信する管理インタフェースと、記憶領域割当要求に従って、計算機に未割当の記憶領域を計算機によりアクセス可能なように割当てた記憶領域割当モジュールと、記憶領域割当要求に従って、計算機と割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当てた性能割当モジュールとを有する。

【0008】

他の実施形態の管理プログラムは、計算機と、計算機によりアクセスされるデータを保持可能な記憶領域と計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジュールとを有する記憶装置とに接続された管理計算機に実行させるためのプログラムである。管理プログラムは、計算機に記憶領域を割当てた要求を受付ける手順と、記憶装置から、記憶装置内部の複数の対象モジュールの実行性能値を受信する手順と、記憶領域割当要求に従って、計算機に未割当の記憶領域を計算機によりアクセス可能なように割当てたように記憶装置に指示する手順と、記憶領域割当要求と対象モジュールの実行性能値とに従って、計算機と割当てた記憶領域との間でのデータアクセスに影響する性能を割当てたように対象モジュールを特定し、かつ、特定した対象モジュールの性能を割当てたように記憶装置に指示する手順とを管理計算機に実行させる。

【0009】

なお、本発明の他の特徴については、本明細書及び添付図面の記載により明

らかにする。

【0010】

【発明の実施の形態】

以下、本発明の実施形態を図を用いて説明する。

●システム構成

図1は、第一の実施形態のシステム構成を示す。図1のシステムは、二台の計算機300（それぞれを区別するときには301と302と称す）をファイバチャネルスイッチ200を介して、記憶装置100に接続している。計算機とファイバチャネルスイッチ、ファイバチャネルスイッチと記憶装置の間の接続には、ファイバチャネル10を用いて接続している。また、本システムには、管理計算機500を設け、記憶装置100の記憶領域の割当と開放を制御させている。管理計算機500と記憶装置100は、専用の通信回線20によって接続している。本発明は、計算機や記憶装置の台数によってなんら制限を受けるものではない。

●計算機

図1に示す計算機300は、プログラムの実行を司るCPU310と、プログラムやプログラムの実行に必要となる情報を格納するメモリ311と、他の計算機との通信を行うためのネットワークインタフェース315と、記憶装置100とデータの交換を行うためのファイバチャネルインタフェース320（以下FCインタフェースと略す）からなる。計算機は300、プログラムの実行によって生成したデータを、FCインタフェース320を介して記憶装置100に割当てた記憶領域に格納することができる。また、計算機300は、プログラムそのものや、プログラムの実行に必要となるデータを、FCインタフェース320を介して記憶装置に割当てられた記憶領域から取得することができる。FCインタフェース320は、WWN（World Wide Name）と呼ばれる識別番号を有している。このWWNは、FCインタフェースごとにユニークな番号が割当てられ、記憶装置は、どのFCインタフェース320が発行した要求かを判断することができる。図1に示したシステムでは、このWWNを計算機300を識別するための識別番号として利用する。WWNは8バイトで表されるが、本実施形態では、便宜

上、計算機 300 に割当てた番号 301 と 302 を用いて説明することにする。

記憶装置

図 1 に示す記憶装置 100 には、二つの FC インタフェース 110（それぞれを区別するときには 111 と 112 と称す）を設けている。FC インタフェース 110 には、帯域制御モジュール 150 を設けている。記憶装置 100 が有する記憶領域は、割当済の記憶領域と、未割当の記憶領域の二つに大別され、記憶領域管理モジュール 120 によって管理する。割当済の記憶領域は、リードライト処理モジュール 140 によって、FC インタフェース 110 に接続され、計算機 300 から利用可能になる。記憶装置 100 には、管理計算機 500 からの記憶領域の割当要求および開放要求を受け取るための要求受取モジュール 122 と、要求受取モジュール 122 が受け取った要求を解釈し実行するための制御モジュール 124 と、割当要求に従って未割当の記憶領域から、必要な記憶領域を取り出して FC インタフェース 110 に接続するための記憶領域割当モジュール 126 と、帯域制御モジュール 150 に帯域制御パラメータを設定するための帯域割当モジュール 128 と、記憶領域にキャッシュを割当てるキャッシュ割当制御モジュール 130 を設けている。帯域制御モジュール 150 は、記憶領域割当要求に従って設定されたパラメータに従って、計算機と割当てた記憶領域との間でのデータアクセスに影響する性能を保証するように、FC インタフェース 110 を制御する性能割当モジュールの一つである。キャッシュ割当制御モジュール 130 は、記憶領域割当要求に従って設定されたパラメータに従って、計算機と割当てた記憶領域との間でのデータアクセスに影響するキャッシュの割当量を制御する性能割当モジュールの一つである。

【0011】

一方、計算機 300 からのデータのリードライト要求は、FC インタフェース 110 が受け取り、リードライト処理モジュール 140 が要求を処理する。リードライト処理モジュール 140 は、記憶領域管理モジュール 120 の管理するマッピングテーブル 121 に基づき、計算機から受け取ったリードライト要求解釈し、指定された記憶領域にデータを格納したり、指定された記憶領域からデータを取得したりする。

【0012】

図2は、記憶装置のハードウェア構成を示す図である。記憶装置100は、プログラムの実行を司るCPU190と、プログラムを格納するメモリ191と、管理計算機500との通信を行うためのネットワークインタフェース195と、FCインタフェース110と、記憶領域を構成する複数台のハードディスク装置180と、ハードディスク装置180に格納するデータや、計算機300から受け取ったデータを一時的に格納するキャッシュ145からなる。記憶領域管理モジュール120、要求受取モジュール122、制御モジュール124、記憶領域割当モジュール126、帯域割当モジュール128、キャッシュ割当制御モジュール130、帯域制御モジュール150は、メモリ191に保持し、CPU190がそのプログラムを実行することによってその機能を実現している。また、後述するマッピングテーブル121、帯域パラメータ算出テーブル129、キャッシュ量算出テーブル131、キャッシュ割当テーブル139、リードライト処理テーブル141などの各モジュールを実行するにあたり必要となる情報も、メモリ191に保持する。

●プログラムの保持

記憶装置100のメモリ191に格納される記憶領域管理モジュール120、要求受取モジュール122、制御モジュール124、記憶領域割当モジュール126、帯域割当モジュール128、キャッシュ割当制御モジュール130、帯域制御モジュール150は、プログラムとして記憶装置100内のROMや磁気ディスク装置等の不揮発記録媒体に記録され、記憶装置の起動時にメモリ191にロードされて実行されるものとする。また、プログラムを記録する媒体は、ROMや磁気ディスク装置以外の他の記憶媒体（CD-ROMやフロッピー（登録商標）ディスク）でも良い。さらに、プログラムは、当該記憶媒体からコンピュータにロードして実行しても良いし、ネットワークを通じて当該記憶媒体にアクセスしてロードするものとしてもよい。

●記憶領域の管理

記憶装置100は、複数台のハードディスク180（本実施形態の場合には3台のハードディスクを用意しておりそれぞれを区別するときには181、182

、183と称す)によって、記憶領域を提供している。本実施形態では、一台のハードディスク180の容量を100GBとして説明する。通常、ハードディスクは、512バイトを一つのブロック(リードライト処理の最小単位)として管理している。100GBのハードディスクの場合、209715200個のブロックを有していることになる。ブロックには、LBA(Logical Block Address)とよばれる番号を割当ててブロックを識別している。100GBのハードディスクの場合、0~209715199番が先頭のブロックから順番に割り振られる。

【0013】

図3~8は、マッピングテーブル121を示す図である。

【0014】

記憶領域管理モジュール120は、図3~8に示すマッピングテーブル121によって、割当済の記憶領域と未割当の記憶領域を、ハードディスク番号とLBA範囲によって管理する。図3の場合、割当済の記憶領域はなく、ハードディスク181、182、183は、全て、未割当状態であることを示す。(図3~8において、ハードディスク番号は図2のそれと一致させて説明している。) また、図4の場合、ハードディスク181のLBA0~41943039は、記憶領域番号91として割当て済みであることを示す。

●アクセス制御と帯域制御の仕組み

次に、記憶装置のアクセス制御と帯域制御の仕組みについて図9を用いて説明する。

【0015】

図9は、アクセス制御と帯域制御を説明するための模式図である。図9では、二台の計算機301と302を、ファイバチャネルスイッチ200を介して、記憶装置100に接続している。記憶装置100には、二つの割当済の記憶領域91と93があり、一つのFCインタフェース111に接続している。この構成では、二台の計算機301と302は、一つのインタフェース111を共有することになる。

【0016】

図10～14は、リードライト処理テーブルを示す図である。リードライト処理モジュール140と帯域制御モジュール150は、図10～14に示すリードライト処理テーブル141を用いて、計算機に対する記憶領域のアクセス制御と帯域制御を行う。アクセス制御はリードライトモジュール140が司り、帯域制御は帯域制御モジュール150が司る。例えば、図13に示すリードライト処理テーブル141によれば、FCインタフェース111には、二つの割当済の記憶領域91と93があり、記憶領域91に対してはWWN301（すなわち計算機301）が、記憶領域93に対してはWWN302（すなわち計算機302）がアクセス可能（リードライト処理可能）であることがわかる。リードライト処理モジュール140は、リードライト処理テーブル141のこの情報に従って、アクセス制御を行う。すなわち、計算機301からのリードライト要求に対しては記憶領域91のみをアクセス可能とし、計算機302からのリードライト要求に対しては記憶領域93のみをアクセス可能とする。また、計算機301と302で共有しているFCインタフェース111で、計算機301からのリードライト要求と、計算機302からのリードライト要求が重なった場合には、帯域制御モジュール150は、リードライト処理テーブル141に保持されたアクセス比率に従い、計算機302と計算機301からのアクセスが競合した場合においても、計算機301からのアクセスが80%、計算機302からのアクセスが10%の割合となるように制御を行う。

●管理計算機

図1に示す管理計算機500は、プログラムの実行を司るCPU510と、プログラムやプログラムの実行に必要となる情報を格納するメモリ511と、プログラムの実行状態を表示するためのディスプレイ515と、管理者の指示を入力するためのキーボード516とマウス517と、記憶装置100に記憶領域の割当要求および開放要求を作成発行するための要求作成発行モジュール520と、記憶装置100と接続するためのネットワークインタフェース590を設けている。

【0017】

図15は、管理計算機の提供する画面を示す図である。管理計算機500上の

ディスプレイ 515 には、図 15 に示す画面を表示し、管理者が記憶領域の作成、削除、および状態管理を行う。図 15 (A) は記憶領域の作成画面 610 である。記憶領域を割当てする計算機を指定する入力フィールド 611、作成する記憶領域の容量を指定する入力フィールド 612、記憶領域を作成するヒントを入力する三つの入力フィールド 613～615、作成を指示する作成ボタン 619 からなる。図 15 (B) は記憶領域の一覧表示および記憶領域の削除を指示する画面 620 である。画面 620 には、計算機ごとに割当済の記憶領域および記憶領域に割当られたヒントを表示する表示領域 621、削除する記憶領域を選択する選択フィールド 622、未割当の記憶領域の容量を表示する表示領域 625、選択フィールド 622 によって選択された記憶領域を開放するための開放ボタン 629 からなる。

●記憶領域の割当要求とヒント

本実施形態では、管理計算機は、DMTF の CIM および WBEM によって規定されるインタフェースと要求のフォーマットに従って、記憶領域の割当要求を記憶装置に発行する。割当要求を発行する際に用いるヒントは、従来技術に示したように、DataAvailabilityHint、AccessRandomnessHint、AccessDirectionHint、AccessSizeHint、AccessLatencyHint、AccessBandwidthWeight、StorageCostHint、StorageEfficiencyHint の八種類に及ぶが、本実施形態では、AccessBandwidthWeight、AccessRandomnessHint、AccessLatencyHint の三つに絞って説明することにする。もちろん、全てのヒントを利用することで、よりよい実施形態になる。また、CIM 以外によって定義されるヒントにおいても適用可能である。

●割当シーケンス

それでは、図を用いながら、本発明に特徴的な記憶領域の割当てと、帯域設定について説明する。ここでは、図 3 および図 10 に示す状態を初期状態と仮定し、すなわち、全ての記憶領域は未割当の状態であることを初期状態として説明する。また、処理の流れについては、図 20～13 のフローチャートを用いて説明する。

【0018】

図20は、ヒントに基づく記憶領域の割当て、性能保証する処理を示すフローチャートである。

【0019】

図21は、記憶領域の割当て処理を示すフローチャートである。

【0020】

図22は、帯域保証する処理のフローチャートである。

●初期化状態から一つ目の記憶領域91の作成

管理者が、管理計算機500を操作して計算機301に記憶領域を割当てることを想定する。管理者は、管理計算機500の表示する画面610を用いて、記憶領域の作成を指示する。ここで管理者は、例えば、

計算機を指定する入力フィールド611に「301」

記憶領域の容量を指定する入力フィールド612に「20GB」

記憶領域を作成するヒントを入力するフィールド613～615にそれぞれ、

AccessBandwidthWeightに「10」

AccessRandomnessHintに「10」

AccessLatencyHint「0」

を入力し(1101)、作成ボタン619を押す(1103)。管理計算機500は、入力フィールドに入力された値を取得し(1105)、CIMに基づく割当要求を作成し(1107)、記憶装置100に発行する(1109)。管理計算機500で作成された割当要求は、管理計算機500のCPU510によって、ネットワークインタフェース590を介して記憶装置100に発行される。記憶装置100の要求受取モジュール122は、ネットワークインタフェース195を介して管理計算機500が発行した割当要求を受け取る(1111)。要求受取モジュール122は、割当要求を受け取ると、制御モジュール124に割当要求を渡す(1113)。制御モジュール124は、要求を解析し、割当要求であることを判定する(1117)。制御モジュール124は、割当要求であることがわかると、まず、作成すべき容量とヒントを取得する(1119)。制御モジュールは、作成すべき容量とヒントを記憶領域割当モジュール126に渡し、記憶領域の割当を依頼する。記憶領域割当モジュール126は、ヒントを元に、指定

された（今の場合には20GB）の記憶領域を、どのハードディスクから割当て
るかをマッピングテーブル121から決定する（1121）。記憶領域を作成す
るハードディスクを選択するアルゴリズムの詳細は、図21にそのフローチャ
ーを示す。図3の状態では、未使用状態であるので（1205）、一番番号の若
いハードディスク181から20GB（LBAの0～41943039）を割当
てる（1213）。記憶領域割当モジュール126は、記憶装置領域番号として
91を割当て（1215）、図4に示すように、マッピングテーブル121を更
新する（1125）。マッピングテーブル121には、ヒントの一つとして受け
取ったAccessBandwidthWeightとAccessRandomnessHint、AccessLatencyHintの値
も保持している。記憶領域割当モジュール126は、記憶領域の割当てが完了す
ると、制御モジュール124に完了を通知する。

【0021】

次に、制御モジュール124は、ヒントを帯域割当モジュール128に渡し、
記憶領域を接続するFCインタフェースの選択と、帯域制御モジュール150に
対して設定するパラメータの算出を依頼する。帯域割当モジュール128は、ヒ
ントから帯域制御モジュール150に対して設定するパラメータを算出する（1
129）。

【0022】

図16は、帯域パラメータ算出テーブルとキャッシュ量算出テーブルを示す図
である。本実施形態では、図16（A）に示す帯域パラメータ算出テーブル12
9を用いてパラメータを求めている。帯域パラメータ算出テーブル129は、0
から10で与えられるAccessBandwidthWeightを帯域制御モジュール150に与
えるパラメータに変換するためのテーブルである。なお、記憶領域を接続するF
Cインタフェースの選択するアルゴリズムのフローチャートは図22を用い詳細
に後述する。前記設定の場合、AccessBandwidthWeightは「10」なので、帯域
制御モジュール150に設定するパラメータは「80%」になる（1301）。
次に、帯域割当モジュール128は、リードライト処理テーブル141を参照し
、帯域を80%割当可能なFCインタフェースを検索する（1303）。図10
に示すように、FCインタフェースに記憶領域は未割当の初期状態としているの

で（1307）、帯域割当モジュール128は、一番番号の若いFCインタフェース111に記憶領域91を接続する（1311）。アクセスを許可する計算機は301なので、リードライト処理テーブル141は、帯域割当モジュール128によって、図11のように更新する（1133）。帯域割当モジュール128は、帯域の割当てを完了すると、制御モジュール124に完了を通知する。

【0023】

次に、制御モジュール124は、ヒントをキャッシュ割当制御モジュール130に渡し、記憶領域91に専用に割当ててるキャッシュ量の選択を依頼する。キャッシュ割当制御モジュール130は、図16（B）に示すキャッシュ量算出テーブル131を用いて記憶領域に割当ててるキャッシュの量を算出する（1135）。キャッシュ量算出テーブル131は、0から10で与えられるAccessLatencyHintとAccessRandomnessHintから記憶領域に専用に割当ててるためのキャッシュ量を算出するためのテーブルである。キャッシュ割当制御モジュール130は、AccessLatencyHintから記憶領域の容量に対するキャッシュのパーセンテージを求め、記憶領域の容量と乗算することによってキャッシュの割当量を算出する。前記ヒントの場合、AccessLatencyHintは「0」なので（1137）、キャッシュ割当制御モジュール130は、記憶領域91には専用のキャッシュを割当てず、制御モジュール124に完了を通知する。

【0024】

制御モジュール124は、要求受取モジュール122に割当要求が完了したことを通知する。要求受取モジュール122は、管理計算機500に割当要求が完了したことを通知する。

【0025】

以上により、管理者が要求した20GBの記憶領域は、記憶装置100に生成され、計算機301から利用できるようになった。以上のように処理することで、ヒントに従って作られた記憶領域91に対して、計算機301と記憶領域91の間の帯域もヒントに基づいて設定することができる。これにより、記憶領域を作成した後に、別のモジュールによって帯域を設定する二度手間が省ける。また、従来のCIMのインタフェースおよび要求のフォーマットになんら変更を加え

ることなく、帯域の設定までを行うことができる。

二つ目の記憶領域 92 の作成

さらに管理者が、管理計算機 500 を操作して計算機 302 に記憶領域を割当ててることを想定する。管理者は、管理計算機 500 の表示する画面 610 を用いて、記憶領域の作成を指示する。ここで管理者は、例えば、

計算機を指定する入力フィールド 611 に「302」

記憶領域の容量を指定する入力フィールド 612 に「20GB」

記憶領域を作成するヒントを入力するフィールド 613～615 にそれぞれ、

AccessBandwidthWeight に「8」

AccessRandomnessHint に「0」

AccessLatencyHint 「0」

を入力し (1101)、作成ボタン 619 を押す (1103)。管理計算機 500 は、入力フィールドに入力された値を取得し (1105)、CIM に基づく割当要求を作成し (1107)、記憶装置 100 に発行する (1109)。管理計算機 500 で作成された割当要求は、先と同様に要求受取モジュール 122 が受け取り (1111) 制御モジュール 124 に渡される (1113)。制御モジュール 124 は、割当要求から、作成すべき容量とヒントを取得する (1119)。制御モジュールは、作成すべき容量とヒントを記憶領域割当モジュール 126 に渡し、記憶領域の割当を依頼する。記憶領域割当モジュール 126 は、ヒントを元に、指定された (今の場合には 20GB) の記憶領域を、どのハードディスクから割当ててるかをマッピングテーブル 121 から決定する (1121)。マッピングテーブル 121 は、図 4 の状態であるので、記憶領域割当モジュール 126 は、ハードディスク 182 から、20GB (LBA の 0～41943039) を割当てて (1213)。ここで、ハードディスク 181 から割当てなかったのは、既に割当済の記憶領域 91 が AccessRandomnessHint が「10」であり、今回作成する記憶領域の AccessRandomnessHint が「0」であるため、シーケンシャルアクセスとランダムアクセスが混在することを避けるためである (1203)。記憶領域割当モジュール 126 は、記憶領域番号として 92 を割当て (1215)、図 5 に示すように、マッピングテーブル 121 を更新する (1125)。先

と同様に、マッピングテーブル 121 には、ヒントとして受けとった AccessBandwidthWeight と AccessRandomnessHint、AccessLatencyHint の値も保持している。記憶領域割当モジュール 126 は、記憶領域の割当てが完了すると、制御モジュール 124 に完了を通知する。

【0026】

次に、制御モジュール 124 は、ヒントを帯域割当モジュール 128 に渡し、記憶領域を接続する FC インタフェースの選択と、帯域制御モジュール 150 に対して設定するパラメータの算出を依頼する。前記設定の場合、AccessBandwidthWeight は「8」なので、帯域制御モジュール 150 に設定するパラメータは「60%」になる(1301)。次に、帯域割当モジュール 128 は、リードライト処理テーブル 141 を参照し、帯域を 60% 割当可能な FC インタフェースを検索する(1303)。図 11 に示すように、FC インタフェース 111 には、既に記憶領域 91 が接続され、帯域を 80% 占有しているので、帯域割当モジュール 128 は、FC インタフェース 112 に記憶領域 92 を接続する(1307, 1311)。アクセスを許可する計算機は 302 なので、リードライト処理テーブル 141 は、帯域割当モジュール 128 によって、図 12 のように更新する(1133)。帯域割当モジュール 128 は、帯域の割当てを完了すると、制御モジュール 124 に完了を通知する。

【0027】

次に、制御モジュール 124 は、ヒントをキャッシュ割当制御モジュール 130 に渡し、記憶領域 92 に専用に割当てるキャッシュ量の選択を依頼する。前記ヒントの場合、AccessLatencyHint は「0」なので、キャッシュ割当制御モジュール 130 は、記憶領域 92 には専用のキャッシュを割当てず(1137)、制御モジュール 124 に完了を通知する。

【0028】

制御モジュール 124 は、要求受取モジュール 122 に割当要求が完了したことを通知する。要求受取モジュール 122 は、管理計算機 500 に割当要求が完了したことを通知する。

【0029】

以上により、管理者が要求した20GBの記憶領域92は、記憶装置100に生成され、計算機302から利用できるようになった。記憶領域92は、ヒントを用いて、記憶領域91とは、異なるハードディスク182の中に作成され、記憶領域91の接続されたFCインタフェース111とは異なるFCインタフェース112に接続することができる。

●三つ目の記憶領域93の作成

さらに管理者が、管理計算機500を操作して計算機302に記憶領域を割当ててることを想定する。管理者は、管理計算機500の表示する画面610を用いて、記憶領域の作成を指示する。ここで管理者は、例えば、

計算機を指定する入力フィールド611に「302」

記憶領域の容量を指定する入力フィールド612に「1GB」

記憶領域を作成するヒントを入力するフィールド613～615にそれぞれ、

AccessBandwidthWeightに「1」

AccessRandomnessHintに「10」

AccessLatencyHint「10」

を入力し(1101)、作成ボタン619を押す(1103)。管理計算機500は、入力フィールドに入力された値を取得し(1105)、CIMに基づく割当要求を作成し(1107)、記憶装置100に発行する(1109)。管理計算機500で作成された割当要求は、先と同様に要求受取モジュール122が受け取り(1111)、制御モジュール124に渡される(1113)。制御モジュール124は、割当要求から、作成すべき容量とヒントを取得する(1119)。制御モジュールは、作成すべき容量とヒントを記憶領域割当モジュール126に渡し、記憶領域の割当を依頼する。記憶領域割当モジュール126は、ヒントを元に、指定された(今の場合には1GB)の記憶領域を、どのハードディスクから割当てるかをマッピングテーブル121から決定する(1121)。マッピングテーブル121は、図5の状態であるので、記憶領域割当モジュール126は、ハードディスク181から、1GB(LBAの41943040～44040191)を割当て(1213)。ここで、ハードディスク181から割当てたのは、既に割当済の記憶領域91がAccessRandomnessHintが「10」であり

、今回作成する記憶領域のAccessRandomnessHintが「10」であるため（1203）、同一ハードディスク内でのアクセスパターンの混在が少なく、かつ、AccessBandwidthWeightが「1」と小さくアクセス頻度も低く、既に割当済の記憶領域91に対する影響が少ないと判断したためである（1211）。ここでは、具体的に同一ハードディスク内に割当てられた記憶領域のAccessBandwidthWeightが15以下になるように制御している（1211）。なお、既に割当てられている記憶領域に対してのアクセスランダム性又はアクセス帯域幅は実行値を測定し、その実行値を相当するAccessRandomnessHint又はAccessBandwidthWeightに変換して比較するようにしても良い。記憶領域割当モジュール126は、記憶領域番号として93を割当て（1215）、図6に示すように、マッピングテーブル121を更新する（1125）。先と同様に、マッピングテーブル121には、ヒントとして受けとったAccessBandwidthWeightとAccessRandomnessHint、AccessLatencyHintの値も保持している。記憶領域割当モジュール126は、記憶領域の割当てが完了すると、制御モジュール124に完了を通知する。

【0030】

次に、制御モジュール124は、ヒントを帯域割当モジュール128に渡し、記憶領域を接続するFCインタフェースの選択と、帯域制御モジュール150に対して設定するパラメータの算出を依頼する。前記設定の場合、AccessBandwidthWeightは「1」なので、帯域制御モジュール150に設定するパラメータは「10%」になる（1301）。次に、帯域割当モジュール128は、リードライト処理テーブル141を参照し、帯域を10%割当可能なFCインタフェースを検索する（1303）。図12に示すように、FCインタフェース111には、既に記憶領域91が接続され、帯域を80%占有している。また、FCインタフェース112には、既に記憶領域92が接続され、帯域を60%占有している。しかし、どちらのFCインタフェースに対しても10%の帯域を割当てることができる。そこで、帯域割当モジュール128は、マッピングテーブル121を参照し（現時点での状態は図6）、記憶領域91と記憶領域92のAccessRandomnessHintの値を確認する（1305）。マッピングテーブル121より、記憶領域91のAccessRandomnessHintの値が、今回作成した記憶領域93のAccessRandom

nessHintの値が同じであることがわかる。そこで、帯域割当モジュール128は、記憶領域91の接続されているFCインタフェース111に記憶領域93を接続する(1311)。アクセスを許可する計算機は302なので、リードライト処理テーブル141は、帯域割当モジュール128によって、図13のように更新する(1133)。帯域割当モジュール128は、帯域の割当てを完了すると、制御モジュール124に完了を通知する。

【0031】

次に、制御モジュール124は、ヒントをキャッシュ割当制御モジュール130に渡し、記憶領域91に専用に割当てるキャッシュ量の選択を依頼する。前記ヒントの場合、AccessLatencyHintは「10」、AccessRandomnessHintは「10」なので、キャッシュ割当制御モジュール130は、キャッシュ割当比率は「100%」となり(1135)、記憶領域93の容量は1GBであるので、キャッシュ割当制御モジュール130は、記憶領域93に割当てるキャッシュ量を1GBと算出する(1139)。

【0032】

図17は、キャッシュ割当テーブルを示す図である。キャッシュ割当制御モジュール93は、図17(A)に示すキャッシュ割当テーブル139を参照し、特定の記憶領域に割当てられていないキャッシュ量が4GBあることを確認する(1141)。そこで、キャッシュ割当制御モジュール130は、記憶領域93に1GBのキャッシュを割当てするため、キャッシュ割当テーブル139を、図17(B)に示すように更新する(1143)。もし、キャッシュ割当テーブルを参照して特定の記憶領域に割当てられていないキャッシュ量が、割当てに必要なキャッシュ量に満たない場合には、制御モジュール124に対して割当てできなかったことを通知することになる。今回の場合には、1GBのキャッシュを割当てることができたので、キャッシュ割当制御モジュール130は、制御モジュール124にキャッシュの割当て完了を通知する。

【0033】

制御モジュール124は、要求受取モジュール122に割当要求が完了したことを通知する。要求受取モジュール122は、管理計算機500に割当要求が完

了したことを通知する。

【0034】

以上により、管理者が要求した1GBの記憶領域93は、記憶装置100に生成され、計算機302から利用できるようになった。記憶領域93は、ヒントを用いて、記憶領域91と同じハードディスク181の中に作成され、記憶領域91の接続されたFCインタフェース111と同じFCインタフェース111に接続された。計算機302からの記憶領域93に対するアクセス頻度が高くなっても、帯域制御モジュール150によって、アクセス比率は10%に抑えられるため、高いアクセス頻度が求められる記憶領域91のアクセスに80%の優先度を保証することができる。また、記憶領域93には、十分なキャッシュを割当ててあるので、リードライトの際の処理時間（レイテンシ）も高速化することができる。

●四つ目の記憶領域94の作成（帯域不足によるエラー）

さらに管理者が、管理計算機500を操作して計算機302に記憶領域を割当ててることを想定する。管理者は、管理計算機500の表示する画面610を用いて、記憶領域の作成を指示する。ここで管理者は、例えば、

計算機を指定する入力フィールド611に「302」

記憶領域の容量を指定する入力フィールド612に「1GB」

記憶領域を作成するヒントを入力するフィールド613～615にそれぞれ、

AccessBandwidthWeightに「10」

AccessRandomnessHintに「0」

AccessLatencyHint「0」

を入力し（1101）、作成ボタン619を押す（1103）。管理計算機500は、入力フィールドに入力された値を取得し（1105）、CIMに基づく割当要求を作成し（1107）、記憶装置100に発行する（1109）。管理計算機500で作成された割当要求は、先と同様に要求受取モジュール122が受け取り（1111）、制御モジュール124に渡される（1113）。制御モジュール124は、割当要求から、作成すべき容量とヒントを取得する（1119）。制御モジュールは、作成すべき容量とヒントを記憶領域割当モジュール12

6に渡し、記憶領域の割当を依頼する。記憶領域割当モジュール126は、ヒントを元に、指定された（今の場合には1GB）の記憶領域を、どのハードディスクから割当ててくるかをマッピングテーブル121から決定する（1121）。マッピングテーブル121は、図6の状態であるので、記憶領域割当モジュール126は、ハードディスク183から、1GB（LBAの0～2097151）を割当てて（1213）。ここで、ハードディスク183から割当てたのは、AccessBandwidthWeightが「10」と大きく、既に割当済の記憶領域91と92への影響を考慮したためである（1203, 1205, 1211）。記憶領域割当モジュール126は、記憶領域番号として94を割当て（1215）、図7に示すように、マッピングテーブル121を更新する（1125）。先と同様に、マッピングテーブル121には、ヒントとして受けとったAccessBandwidthWeightとAccessRandomnessHint、AccessLatencyHintの値も保持している。記憶領域割当モジュール126は、記憶領域の割当てが完了すると、制御モジュール124に完了を通知する。

【0035】

次に、制御モジュール124は、ヒントを帯域割当モジュール128に渡し、記憶領域を接続するFCインタフェースの選択と、帯域制御モジュール150に対して設定するパラメータの算出を依頼する。前記設定の場合、AccessBandwidthWeightは「10」なので、帯域制御モジュール150に設定するパラメータは「80%」になる（1301）。次に、帯域割当モジュール128は、リードライト処理テーブル141を参照し、帯域を80%割当可能なFCインタフェースを検索する（1303）。図13に示すように、FCインタフェース111およびFCインタフェース112とも、新たに帯域を80%割当てることができない（1307）。帯域割当モジュール128は、帯域を割当られなかったことを制御モジュール124に通知する（1399）。制御モジュール124は、帯域の割当てが失敗したことの通知を受けると（1129）、記憶領域割当モジュール126に対して記憶領域94の割当を取り消すよう依頼する（1131）。記憶領域割当モジュール126は、マッピングテーブル121を図6の状態に戻す。制御モジュール124は、要求受取モジュール122に割当要求が失敗したこと

を通知する。要求受取モジュール 122 は、管理計算機 500 に割当要求が失敗したことを通知する。

【0036】

以上により、管理者が要求した 1GB の記憶領域は、記憶装置 100 に生成されない。管理者は、ヒントを再設定して再度作成ボタンを押すことができる。また、制御モジュール 124 は、要求受取モジュール 122 に割当要求が失敗したことを通知する際に、最大割当可能な AccessBandwidthWeight の値（今回の場合には FC インタフェース 112 の残り 40% が最大なので図 16 (A) に示す帯域パラメータ算出テーブル 129 から逆引きすると「6」とわかる）「6」を返し、要求受取モジュールは、この AccessBandwidthWeight の値「6」を管理計算機 500 に通知することで、管理者に AccessBandwidthWeight に最大設定可能な値が「6」であることを教えることも良い。

●開放シーケンス

管理者が、管理計算機 500 を操作して計算機 301 の記憶領域 91 を開放することを想定する。管理者は、管理計算機 500 の表示する画面 620 を用いて、記憶領域 91 の開放を指示する。ここで管理者は、画面 620 上の記憶領域 91 に対応する選択フィールド 622 をチェックし、開放ボタン 629 を押す。管理計算機 500 は、選択フィールドで選択された記憶領域の番号「91」を取得し、CIM に基づく開放要求を作成し、記憶装置 100 に発行する。管理計算機 500 で作成された開放要求は、管理計算機 500 の CPU 510 によって、ネットワークインタフェース 590 を介して記憶装置 100 に発行される。記憶装置 100 の要求受取モジュール 122 は、ネットワークインタフェース 195 を介して管理計算機 500 が発行した開放要求を受け取る。要求受取モジュール 122 は、開放要求を受け取ると、制御モジュール 124 に開放要求を渡す。制御モジュール 124 は、要求を解析し、開放要求であることを判定する。制御モジュール 124 は、帯域割当モジュール 126 に記憶領域 91 の占有している帯域の開放を依頼する。帯域割当モジュール 128 は、記憶領域 91 に関する情報をリードライト制御テーブル 141 から削除する。リードライト制御テーブル 141 は、図 14 のように更新される。次に、制御モジュール 124 は、記憶領域割

当モジュール 126 に記憶領域 91 が占有しているブロックの開放を依頼する。記憶領域割当モジュール 125 は、記憶領域 91 に関する情報をマッピングテーブル 121 から削除する。マッピングテーブル 121 は、図 8 のように更新される。帯域割当モジュール 128 における帯域の開放処理と、記憶領域割当モジュールにおけるブロックの開放処理が終了すると、制御モジュール 124 は、要求受取モジュール 122 に開放要求が完了したことを通知する。要求受取モジュール 122 は、管理計算機 500 に開放要求が完了したことを通知する。管理計算機 500 は、開放要求の完了を受け取ると、画面 620 を更新する。

●システム構成

図 18 は、第二の実施形態のシステム構成を示す。図 18 のシステムも、図 1 のシステムと同様に、二台の計算機 300 をファイバチャネルスイッチ 200 を介して、記憶装置 100 に接続している。計算機とファイバチャネルスイッチ、ファイバチャネルスイッチと記憶装置の間の接続には、ファイバチャネル 10 を用いて接続している。また、本システムには、管理計算機 500 に加えて、第二の管理計算機 400 を設け、記憶装置 100 の記憶領域の割当と開放を制御させている。管理計算機 500 と第二の管理計算機 400、第二の管理計算機 400 と記憶装置 100 は、専用の通信回線 20 によって接続している。

●第二の管理計算機

図 18 に示す第二の管理計算機 400 は、プログラムの実行を司る CPU 410 と、プログラムやプログラムの実行に必要となる情報を格納するメモリ 411 と、管理計算機 500 と接続するためのネットワークインタフェース 490 と、記憶装置 100 と接続するためのネットワークインタフェース 495 を設けている。第二の管理計算機 400 には、管理計算機 500 からの割当要求および開放要求を受け取るための要求受取モジュール 122、制御モジュール 124、帯域割当モジュール 128、キャッシュ割当制御モジュール 130 を設けている。第二の管理計算機に設けた制御モジュール 124、帯域割当モジュール 128、キャッシュ割当制御モジュール 130 は、それぞれ、記憶装置 100 に設けた記憶領域管理モジュール 120、帯域制御モジュール 150、キャッシュ 145 と、ネットワークインタフェース 495 およびネットワークインタフェース 195 を

介して通信しながらその処理を実行する。本実施形態では、要求受取モジュール 122、制御モジュール 124、帯域割当モジュール 128、キャッシュ割当制御モジュール 130 を第二の管理計算機 400 に設ける形態で記載したが、管理計算機 500 に設けても実現することができる。

●プログラムの保持

第二の管理計算機 400 のメモリ 411 に格納される要求受取モジュール 122、制御モジュール 124、帯域割当モジュール 128、キャッシュ割当制御モジュール 130 は、プログラムとして CD-ROM 等の記録媒体に記録され、磁気ディスク等に格納された後、メモリにロードされて実行されるものとする。また、プログラムを記録する媒体は、CD-ROM 以外の他の記憶媒体でも良い。さらに、プログラムは、当該記憶媒体からコンピュータにインストールしても良いし、ネットワークを通じて当該記憶媒体にアクセスしてプログラムを使用するものとしてもよい。

●割当シーケンス

記憶領域の割当処理に関しては、基本的には、第一の実施形態で示した処理をそのまま当てはめることができる。本実施形態に特徴的な処理としては、第二の管理計算機 400 に設けた各モジュールと記憶装置 100 に設けた各モジュールが、ネットワークインタフェース 495 およびネットワークインタフェース 195 を介して通信しながらその処理を実行する。

●システム構成

図 19 は、第三の実施形態のシステム構成を示す。図 19 のシステムでは、二台の計算機 300 を二台の IP ルータ 220 を介して、記憶装置 700 に接続している。計算機と IP ルータ、IP ルータ間、IP ルータと記憶装置の間の接続には、IP ネットワーク 30 を用いて接続している。このため、計算機には、ファイバチャネルインタフェース 320 に代えて、IP インタフェース 370 を設けている。また、第一の実施形態と同様、本システムにも、管理計算機 500 を設け、記憶装置 700 の記憶領域の割当と開放を制御させている。管理計算機 500 と記憶装置 700 は、専用の通信回線 20 によって接続している。

帯域制御の仕組み

IP ルータにおける帯域制御の仕組みは、RSVPとして知られている。IP ルータ 200 には、RSVP を設定するための帯域制御モジュール 229 を設けている。

●記憶装置

図 19 に示す記憶装置 700 には、二つの IP インタフェース 710 を設けている。IP インタフェース 710 と FC インタフェース 110 の間には、ネットワークファイルシステム処理モジュール 720 を設け、IP インタフェース 710 によって受け取るリードライト要求を、FC インタフェース 110 が解釈できるリードライト要求に変換する。FC インタフェース 110 から先のリードライト処理については、第一の実施形態と同様である。ただし、記憶装置 700 には、帯域制御モジュールを設けておらず、IP ルータ 220 が有する帯域制御モジュール 229 を利用する。よって、記憶装置 700 には、IP ルータ 220 の帯域制御モジュール 229 と通信するためのネットワークインタフェース 199 と、IP ルータ 220 の帯域制御モジュール 229 に、帯域割当モジュール 128 が算出した帯域制御パラメータを設定するための IP ルータ帯域設定モジュール 160 を設けた。IP ルータ帯域設定モジュール 160 は、帯域割当モジュール 128 が算出したパラメータを、IP ルータ 220 の RSVP パラメータに変換して帯域設定を行う。

●割当シーケンス

記憶領域の割当処理に関しては、基本的には、第一の実施形態で示した処理をそのまま当てはめることができる。本実施形態に特徴的な処理としては、記憶装置 700 の帯域割当モジュール 128 が、算出したパラメータを設定する際に、IP ルータ帯域設定モジュール 160 にそのパラメータを渡し、IP ルータ帯域設定モジュール 160 は、ネットワークインタフェース 199 を介して、IP ルータ 220 に RSVP パラメータを設定する。

●システム構成

図 23 は、第四の実施形態のシステム構成を示す。図 23 のシステムは、第二の実施形態で説明した図 18 と同様に第二の管理計算機 400 を設け、二台の記憶装置 100 と 101 の記憶領域の割当と開放を制御させている。第二の実施形

態では、第二の管理計算機 400 に、帯域割当モジュール 128 とキャッシュ割当制御モジュール 130 を設けていたが、本実施形態では、記憶装置に設けている。記憶装置 101 は、基本的に記憶装置 100 と同様の構成であるが、記憶装置 100 に設けている FC インタフェース 110 (111 と 112) に比べ、より高速な転送が可能な FC インタフェース 115 (116 と 117) を設けている。FC インタフェースには、1 Gbps と 2 Gbps の仕様のものがあり、ここでは、FC インタフェース 110 の性能を 1 Gbps、FC インタフェース 115 の性能を 2 Gbps として説明する。

● インタフェース性能通知モジュール

第二の管理計算機 400 の制御モジュール 124 は、それぞれの記憶装置が有する FC インタフェースの性能を把握する必要がある。このため、本実施形態では、記憶装置 100 および 101 にインタフェース性能通知モジュール 170 を設けた。インタフェース性能通知モジュール 170 は、第二の管理計算機 400 の制御モジュール 124 からの性能問い合わせ要求に従って、インタフェースの性能情報を応答する。なお、インタフェース性能通知モジュール 170 は、定期的に制御モジュール 124 へ性能情報を送信するようにしてもよい。また、性能情報として、キャッシュの使用率、各記憶領域自体のアクセス性能等を含めて、記憶領域、キャッシュを割当て際の基準としてもよい。

【0037】

図 24 は、性能情報を示す図である。本実施形態の場合、記憶装置 100 のインタフェース性能通知モジュール 170 からは、図 24 (A) に示すような性能情報 171 が返却され、1 Gbps の FC インタフェース 111 と 112 があることがわかる。同様に、記憶装置 101 のインタフェース性能通知モジュール 170 からは、図 24 (B) に示すような性能情報 172 が返却され、2 Gbps の FC インタフェース 116 と 117 があることがわかる。第二の管理計算機 400 の制御モジュール 124 は、メモリ 411 にこれを保持する。第二の管理計算機 400 の制御モジュール 124 は、第二の管理計算機の起動時や、新しい記憶装置を検出した場合、管理計算機 500 から収集要求受信時などに、性能問い合わせ要求を発行し、FC インタフェースの性能情報を収集する。

プログラムの保持

本実施形態で新たに設けたインタフェース性能通知モジュール170は、プログラムとして記憶装置(100および101)内のROMや磁気ディスク装置等の不揮発記録媒体に記録され、記憶装置の起動時にメモリ191にロードされて実行されるものとする。また、プログラムを記録する媒体は、ROMや磁気ディスク装置以外の他の記憶媒体(CD-ROMやフロッピー(登録商標)ディスク)でも良い。さらに、プログラムは、当該記憶媒体からコンピュータにロードして実行しても良いし、ネットワークを通じて当該記憶媒体にアクセスしてロードするものとしてもよい。

●第四の実施形態における帯域パラメータ算出テーブル

図25は、本実施形態で使用する、帯域パラメータ算出テーブル129を示す。図16に示した帯域パラメータ算出テーブル129との違いは、帯域制御モジュールに設定するパラメータが、1Gbpsの場合と、2Gbpsの場合の、二つの場合によって個別に用意されている点である。帯域パラメータ算出テーブル129中に示す「*」は、1GbpsのFCインタフェースでは、「ヒント」を満たす帯域の割当てができないことを示している。たとえば、AccessBandwidthWeightが「10」の場合、2GbpsのFCインタフェースでは、帯域設定を80%とすることで、受け取った「ヒント」を満たすことができるが、1GbpsのFCインタフェースでは、「ヒント」を満たすことができないため、作成した記憶領域を1GbpsのFCインタフェースに接続することは行えない。本実施形態では、二種類の性能の違いについて説明したが、さらに10Gbps等のFCインタフェースが混在した場合にも、同様に帯域パラメータ算出テーブルを作成することで対応することができる。

●割当てシーケンス

図26は、第四の実施形態における割当て処理のフローチャートである。

【0038】

それでは、図を用いながら、本実施形態に特徴的な記憶領域の割当てと、FCインタフェースの選択、そして、帯域設定について説明する。第一の実施形態との違いは、記憶領域を作成する際に、「ヒント」から必要とされる帯域を求めた

場合に、記憶領域を接続することのできない F C インタフェースを有する記憶装置が存在する場合は考えられる点にある。この処理を行うため、第二の管理計算機 4 0 0 は、各記憶装置から収集した F C インタフェースの性能情報（1 7 1 と 1 7 2）、帯域パラメータ算出テーブル 1 2 9 を保持している。また処理の流れについては、第二の管理計算機 4 0 0 の制御モジュール 1 2 4 の処理を中心に図 2 6 のフローチャートを用いて説明する。

●高速な記憶領域の作成（高速な F C インタフェースの選択）

管理者が、管理計算機 5 0 0 を操作して計算機 3 0 1 に記憶領域を割当ててることを想定する。管理者は、管理計算機 5 0 0 の表示する画面 6 1 0 を用いて、記憶領域の作成を指示する。ここで管理者は、例えば、計算機を指定する入力フィールド 6 1 1 に「3 0 1」記憶領域の容量を指定する入力フィールド 6 1 2 に「1 G B」記憶領域を作成するヒントを入力するフィールド 6 1 3 ～ 6 1 5 にそれぞれ、

AccessBandwidthWeight に「1 0」

AccessRandomnessHint に「0」

AccessLatencyHint 「0」

を入力し、作成ボタン 6 1 9 を押す。管理計算機 5 0 0 は、入力フィールドに入力された値を取得し、C I M に基づく割当て要求を作成し、第二の管理計算機 4 0 0 に発行する。第二の管理計算機 4 0 0 の要求受取モジュール 1 2 2 は、割当て要求を受け取ると、制御モジュール 1 2 4 に割当て要求を渡す。制御モジュール 1 2 4 は、要求を解析し、割当て要求であることを判定する（1 7 0 1）。制御モジュール 1 2 4 は、割当て要求であるとわかると、まず、作成すべき容量とヒントを取得する（1 7 0 3）。制御モジュール 1 2 4 は、ヒントのひとつとして受け取った AccessBandwidthWeight を元に、帯域パラメータ算出テーブル 1 2 9 を参照し、設定すべきパラメータと、そのパラメータを設定することのできる F C インタフェースの種別を取得する（1 7 0 5）。今回の場合、AccessBandwidthWeight は「1 0」なので、2 G b p s の F C インタフェースに「8 0 %」を設定しなければならない。すなわち、1 G b p s の F C インタフェースに記憶領域を接続しては、ヒントを満足することができない。帯域パラメータ算出テーブル 1 2 9

の参照結果と、性能情報（171, 172）を元に、第二の管理計算機400の制御モジュール124は、記憶装置101の帯域割当モジュール128に、80%の帯域の割当てが可能なFCインタフェースの検索を依頼する（1707）。帯域割当モジュール128は、帯域の割当て可能なFCインタフェースがあることを確認すると、帯域制御モジュールにその帯域パラメータを設定し、そのFCインタフェース番号とそのFCインタフェースの残帯域を制御モジュール124に返却する（1709）。次に、帯域が割当て可能なFCインタフェースがある場合（1711）、制御モジュール124は、作成する容量（この場合は1GB）と、FCインタフェース番号、AccessRandomnessHintを指定し、記憶領域割当モジュール126に記憶領域の作成と、作成した記憶領域の所定のFCインタフェースへの接続を依頼する（1713）。記憶領域割当モジュール126は、要求に合致する記憶領域を作成可能な場合には、記憶領域を作成し、指定されたFCインタフェースに作成した記憶領域を接続し、第二の管理計算機400の制御モジュール124に処理の完了を通知する（1715）。第二の管理計算機400の制御モジュール124は、管理計算機500に処理の完了を通知する（1727）。

●低速な記憶領域の作成

次にAccessBandwidthWeightに「1」を与えられた場合を説明する。この場合、前記と同様に、第二の管理計算機400は、帯域パラメータ算出テーブル129を参照し、設定すべきパラメータと、そのパラメータを設定することのできるFCインタフェースの種別を取得する（1705）。今回の場合、AccessBandwidthWeightは「1」なので、1GbpsのFCインタフェースの場合には「20%」を、2Gbpsのインタフェースの場合には「10%」を設定することのできるFCインタフェースを確保する必要がある。第二の管理計算機400は、帯域の割当てが可能なFCインタフェースを検索するように各記憶装置に依頼する（1707）。記憶装置100に対しては、1GbpsのFCインタフェースを有するので「20%」を、記憶装置101に対しては、2GbpsのFCインタフェースを有するので「10%」をそれぞれ指定する。各記憶装置の帯域割当モジュール128は、帯域の割当て可能なFCインタフェースがあることを確認す

ると、帯域制御モジュールにその帯域パラメータを設定し、そのFCインタフェース番号とそのFCインタフェースの残帯域を制御モジュール124に返却する(1709)。次に、帯域を割当て可能なFCインタフェースを有する記憶装置それぞれに対して、作成する容量とFCインタフェース番号を指定し、記憶領域割当モジュール126に記憶領域の作成と、作成した記憶領域の所定のFCインタフェースへの接続を依頼する(1713)。複数の記憶装置で記憶領域の作成が成功した場合(1719)には、FCインタフェースの残帯域がもっとも少ないFCインタフェースに接続した記憶領域を残して、それぞれの記憶領域割当モジュール126に対して記憶領域は開放するよう依頼する(1721)。また、使用しなかった記憶領域が接続されていたFCインタフェースの帯域についてもそれぞれの帯域割当モジュール128に対して開放するよう依頼する(1723)。開放処理が完了すると、第二の管理計算機400の制御モジュール124は、管理計算機500に処理の完了を通知する(1727)。

●FCインタフェースの選択のアルゴリズム

先に「残帯域が最も少ない」と記載した点について補足する。例えば、図12に示すような状態では、記憶領域91のためにFCインタフェース111の帯域は既に80%割当済みで、記憶領域92のためにFCインタフェース112の帯域は60%割当済みである。ここで、記憶領域93のために帯域を10%割当てする場合、FCインタフェース111と112の二つの選択肢があるが、図13のように、FCインタフェース111から帯域を10%割当てること、FCインタフェース111の残帯域は10%、FCインタフェース112の残帯域は40%となる。逆に、FCインタフェース112から帯域を10%割当てると、FCインタフェース111の残帯域は20%、FCインタフェース112の残帯域は30%となる。この場合、前者の場合、すなわち、FCインタフェース111に帯域を10%割当てた場合のほうが、未割当の連続する帯域が40%(FCインタフェース112の残帯域)と大きくなる。未割当の連続する帯域が大きいほど、当然ながら、記憶領域の割当処理時に帯域が割当てられずにエラーとなるケースを減らすことができる。

【0039】

以上記載のように処理することで、記憶領域を作成する際に指定するヒントを用いて、記憶領域の割当を行うだけでなく、計算機と記憶領域の間の帯域をも設定したり、記憶領域に割当てたキャッシュの量を設定することで、従来、記憶領域の作成とは別に行っていた設定を一度に行うことができる。よって、管理者には、記憶領域を作成する際に指定したヒントに対する帯域設定のパラメータやキャッシュ割当量を選択するノウハウが不要になり、高度な知識が要求されることもない。また、帯域設定のパラメータの選択を誤る危険性もなく、帯域設定のパラメータの設定ミスもなくすることができる。

【0040】

さらに、従来のCIMのインタフェースおよび要求のフォーマットになんら変更を加えることなく、帯域の設定やキャッシュの割当量の設定を行うことができる。

【0041】

【発明の効果】

本発明によると、記憶領域を割当て際に指定する記憶領域に対する領域割当要求指針（ヒント）を用いて記憶領域を割当てるとともに、このヒントを用いて帯域制御等の性能保証をも実行することにより、管理者による割当てた記憶領域に対する計算機からのアクセス性能を保証するための設定を自動化することができる。

【図面の簡単な説明】

- 【図1】 第一の実施形態におけるコンピュータシステムの構成を示す図
- 【図2】 記憶装置のハードウェア構成を示す図
- 【図3】 マッピングテーブルAを示す図
- 【図4】 マッピングテーブルBを示す図
- 【図5】 マッピングテーブルCを示す図
- 【図6】 マッピングテーブルAを示す図
- 【図7】 マッピングテーブルAを示す図
- 【図8】 マッピングテーブルAを示す図
- 【図9】 アクセス制御と帯域制御を説明するための模式図

- 【図 10】 リードライト処理テーブルAを示す図
- 【図 11】 リードライト処理テーブルBを示す図
- 【図 12】 リードライト処理テーブルCを示す図
- 【図 13】 リードライト処理テーブルDを示す図
- 【図 14】 リードライト処理テーブルEを示す図
- 【図 15】 管理計算機の提供する画面を示す図
- 【図 16】 帯域パラメータ算出テーブルとキャッシュ量算出テーブルを示す

図

- 【図 17】 キャッシュ割当テーブルを示す図
- 【図 18】 第二の実施形態におけるコンピュータシステムの構成を示す図
- 【図 19】 第三の実施形態におけるコンピュータシステムの構成を示す図
- 【図 20】 ヒントに基づく記憶領域、帯域、キャッシュの割当て処理を示す

フローチャート図

- 【図 21】 記憶領域の割当て処理を示すフローチャート図
- 【図 22】 帯域の割当て処理を示すフローチャート図
- 【図 23】 第四の実施形態におけるコンピュータシステムの構成を示す図
- 【図 24】 性能情報を示す図
- 【図 25】 第四の実施形態における帯域パラメータ算出テーブルを示す図
- 【図 26】 第四の実施形態における割当て処理のフローチャート図

【符号の説明】

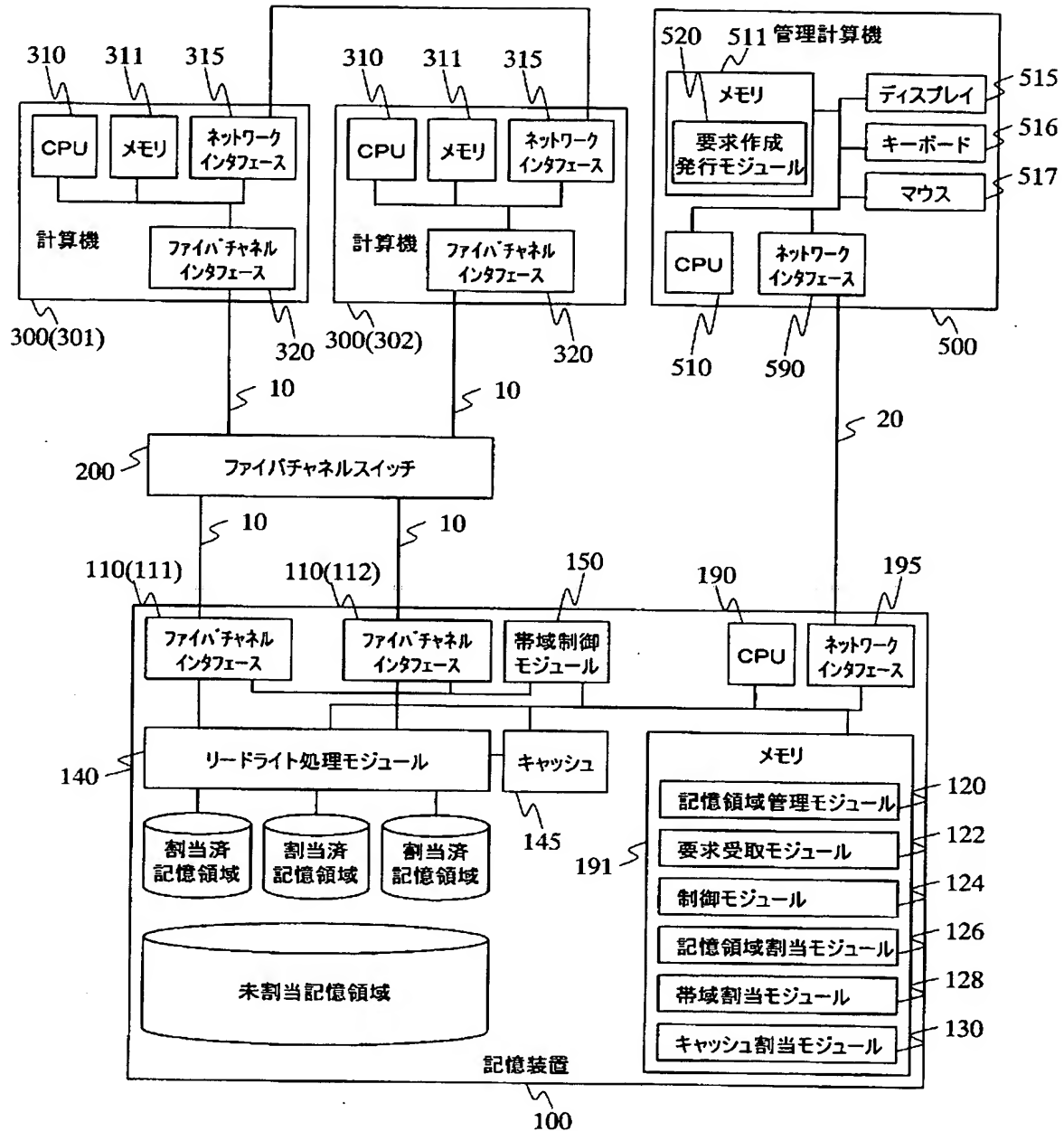
10…ファイバチャネル, 20…通信回線, 30…IPネットワーク, 91…記憶領域, 92…記憶領域, 93…記憶領域, 94…記憶領域, 100…記憶装置, 110…ファイバチャネルインタフェース, 120…記憶領域管理モジュール, 121…マッピングテーブル, 122…要求受取モジュール, 124…制御モジュール, 126…記憶領域割当モジュール, 128…帯域割当モジュール, 129…帯域パラメータ算出テーブル, 130…キャッシュ割当制御モジュール, 131…キャッシュ量算出テーブル, 139…キャッシュ割当テーブル, 140…リードライト処理モジュール, 141…リードライト処理テーブル, 145…キャッシュ, 150…帯域制御モジュール, 160…IPルータ帯域設定モジ

ュール, 180…ハードディスク, 190…CPU, 191…メモリ, 195…
ネットワークインタフェース, 199…ネットワークインタフェース, 200…
ファイバチャネルスイッチ, 220…IPルータ, 229…帯域制御モジュール
, 300…計算機, 310…CPU, 311…メモリ, 315…ネットワークイ
ンタフェース, 320…ファイバチャネルインタフェース, 370…IPインタ
フェース, 400…第二の管理計算機, 410…CPU, 411…メモリ, 49
0…ネットワークインタフェース, 495…ネットワークインタフェース, 50
0…管理計算機, 510…CPU, 511…メモリ, 515…ディスプレイ, 5
16…キーボード, 517…マウス, 520…要求作成発行モジュール, 590
…ネットワークインタフェース, 610…記憶領域の作成画面, 620…記憶領
域の一覧表示および記憶領域の削除を指示する画面, 700…記憶装置, 710
…IPインタフェース, 720…ネットワークファイルシステム処理モジュール

【書類名】 図面

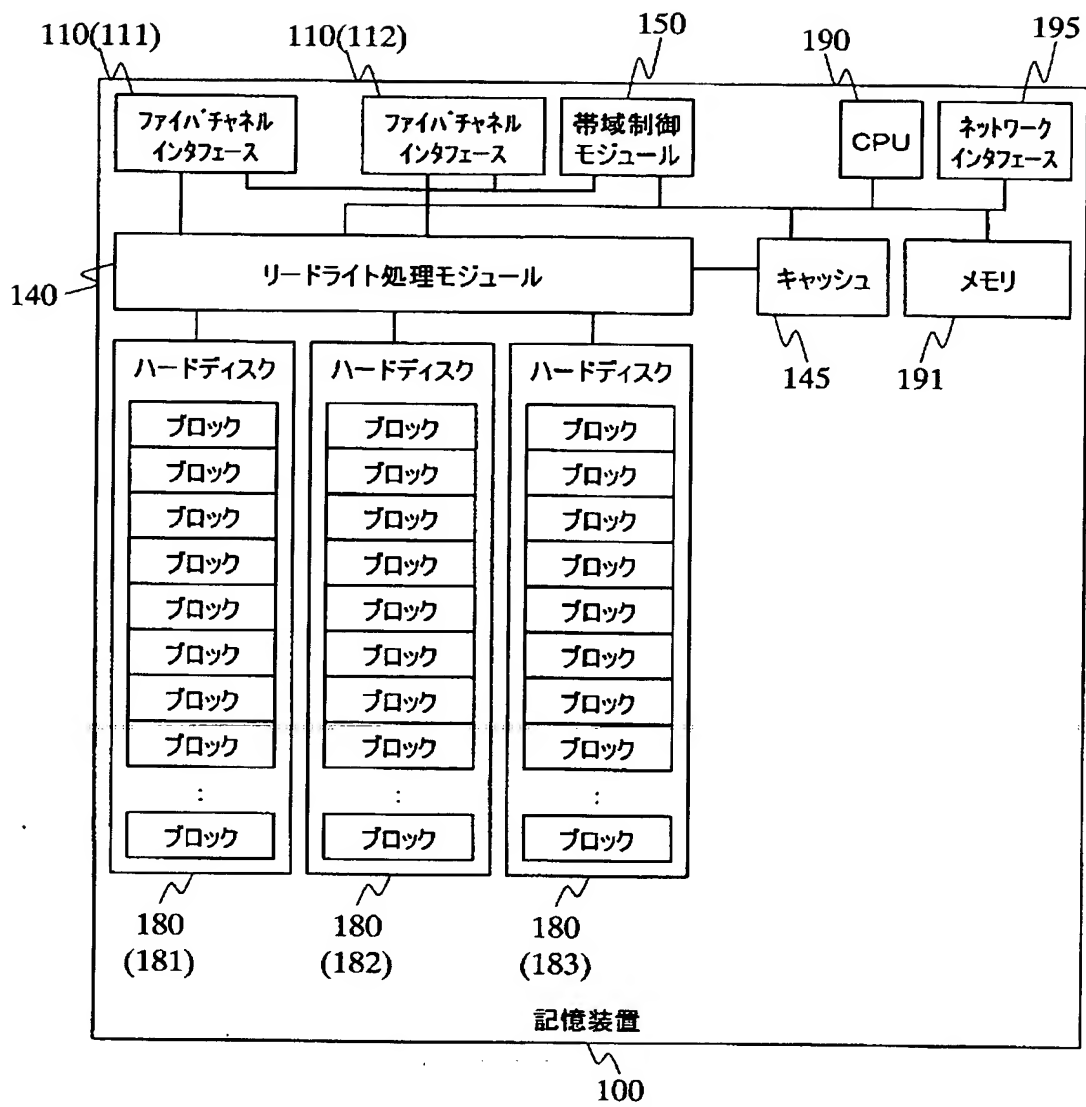
【図1】

図1



【図 2】

図2



【図 3】

図3

121
N

	ハードディスク 番号	LBA範囲	ヒント			記憶領域 番号
			ABW	ARH	ALH	
割当済 記憶領域						
未割当 記憶領域	181	0~209715199				
	182	0~209715199				
	183	0~209715199				

ABW: Access Bandwidth Weight
 ARH: Access Randomness Hint
 ALH: Access Latency Hint

【図 4】

図4

121
N

	ハードディスク 番号	LBA範囲	ヒント			記憶領域 番号
			ABW	ARH	ALH	
割当済 記憶領域	181	0~41943039	10	10	0	91
未割当 記憶領域	181	41943040~ 209715199				
	182	0~ 209715199				
	183	0~ 209715199				

ABW: Access Bandwidth Weight
 ARH: Access Randomness Hint
 ALH: Access Latency Hint

【図 5】

図5

121
~

	ハードディスク 番号	LBA範囲	ヒント			記憶領域 番号
			ABW	ARH	ALH	
割当済 記憶領域	181	0~ 41943039	10	10	0	91
	182	0~ 41943039	8	0	0	92
未割当 記憶領域	181	41943040~ 209715199				
	182	41943040~ 209715199				
	183	0~ 209715199				

ABW: Access Bandwidth Weight

ARH: Access Randomness Hint

ALH: Access Latency Hint

【図 6】

図6

121
~

	ハードディスク 番号	LBA範囲	ヒント			記憶領域 番号
			ABW	ARH	ALH	
割当済 記憶領域	181	0~ 41943039	10	10	0	91
	182	0~ 41943039	8	0	0	92
	181	41943040~ 44040191	1	10	10	93
未割当 記憶領域	181	44040192~ 209715199				
	182	41943040~ 209715199				
	183	0~ 209715199				

ABW: Access Bandwidth Weight

ARH: Access Randomness Hint

ALH: Access Latency Hint

【図 7】

図7

121
~

	ハードディスク 番号	LBA範囲	ヒント			記憶領域 番号
			ABW	ARH	ALH	
割当済 記憶領域	181	0~ 41943039	10	10	0	91
	182	0~ 41943039	8	0	0	92
	181	41943040~ 44040191	1	10	10	93
	183	0~ 2097151	10	0	0	94
未割当 記憶領域	181	44040192~ 209715199				
	182	41943040~ 209715199				
	183	2097152~ 209715199				

ABW: Access Bandwidth Weight
 ARH: Access Randomness Hint
 ALH: Access Latency Hint

【図 8】

図8

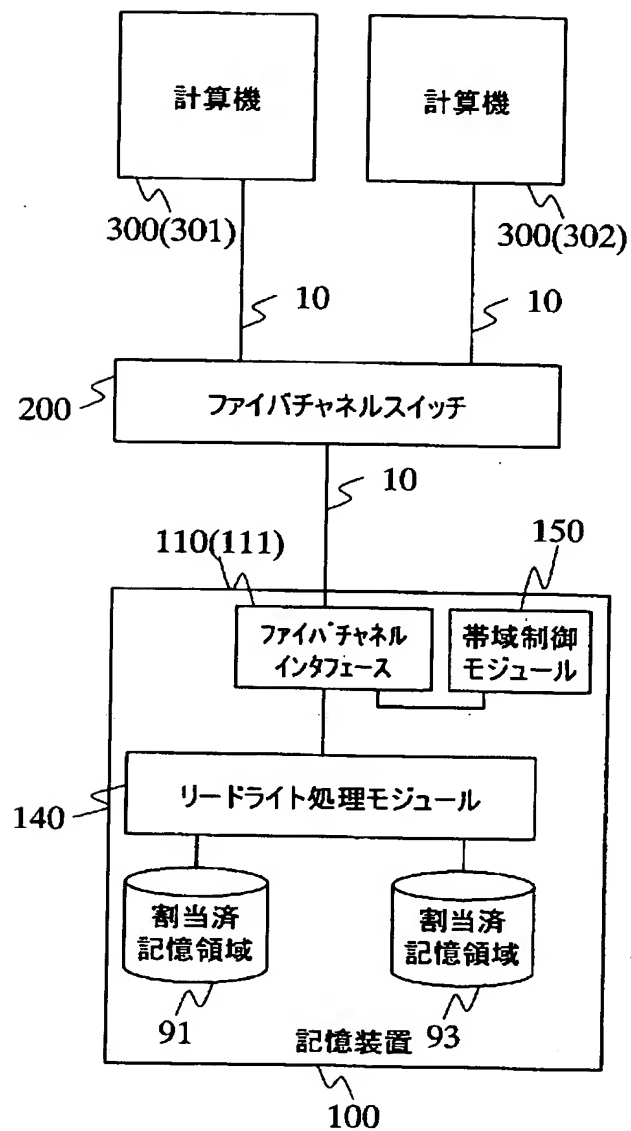
121
~

	ハードディスク 番号	LBA範囲	ヒント			記憶領域 番号
			ABW	ARH	ALH	
割当済 記憶領域	182	0~ 41943039	8	0	0	92
	181	41943040~ 44040191	1	10	10	93
	183	0~ 2097151	10	0	0	94
未割当 記憶領域	181	44040192~ 209715199				
	182	41943040~ 209715199				
	183	0~ 209715199				

ABW: Access Bandwidth Weight
 ARH: Access Randomness Hint
 ALH: Access Latency Hint

【図9】

図9



【図 10】

図10

141
~

FCインターフェース 番号	記憶領域 番号	アクセス可能な計算機の WWN	アクセス比率 (帯域)
111			
112			

【図 11】

図11

141
~

FCインターフェース 番号	記憶領域 番号	アクセス可能な計算機の WWN	アクセス比率 (帯域)
111	91	301	80%
112			

【図 12】

図12

141

FCインターフェース 番号	記憶領域 番号	アクセス可能な計算機の WWN	アクセス比率 (帯域)
111	91	301	80%
112	92	302	60%

【図 13】

図13

141

FCインターフェース 番号	記憶領域 番号	アクセス可能な計算機の WWN	アクセス比率 (帯域)
111	91	301	80%
	93	302	10%
112	92	302	60%

【図 14】

図14

-141
~

FCインターフェース 番号	記憶領域 番号	アクセス可能な計算機の WWN	アクセス比率 (帯域)
111	93	302	10%
112	92	302	60%

【図 15】

図15
(A)

610

記憶領域の作成画面

計算機: 611

容量: 612

ヒント:

Access Bandwidth Weight: 613

Access Randomness Hint: 614

Access Latency Hint: 615

作成 619

↑

↓

(B)

620

記憶領域の状態表示および削除指示

622

621

計算機	選択 フィールド	記憶 領域 番号	ヒント		
			Access Bandwidth Weight	Access Randomness Hint	Access Latency Hint
301	✓	91	10	10	0
302		92	8	0	0
		93	1	10	10

625

開放 619

↑

↓

【図 16】

図16
(A)

129
N

Access Bandwidth Weight	帯域制御モジュール に設定する パラメータ
0	5%
1	10%
2	15%
3	20%
4	25%
5	30%
6	40%
7	50%
8	60%
9	70%
10	80%

(B)

131
N

AccessRandomness Hint AccessLatencyHint	専用キャッシュ割当率	
	0~5	6~10
0	0%	0%
1	1%	10%
2	2%	20%
3	3%	30%
4	4%	40%
5	5%	50%
6	6%	60%
7	7%	70%
8	8%	80%
9	9%	90%
10	10%	100%

【図 17】

図17
(A)

139

	記憶領域 番号	キャッシュ容量
割当済 キャッシュ		
未割当 キャッシュ		4GB

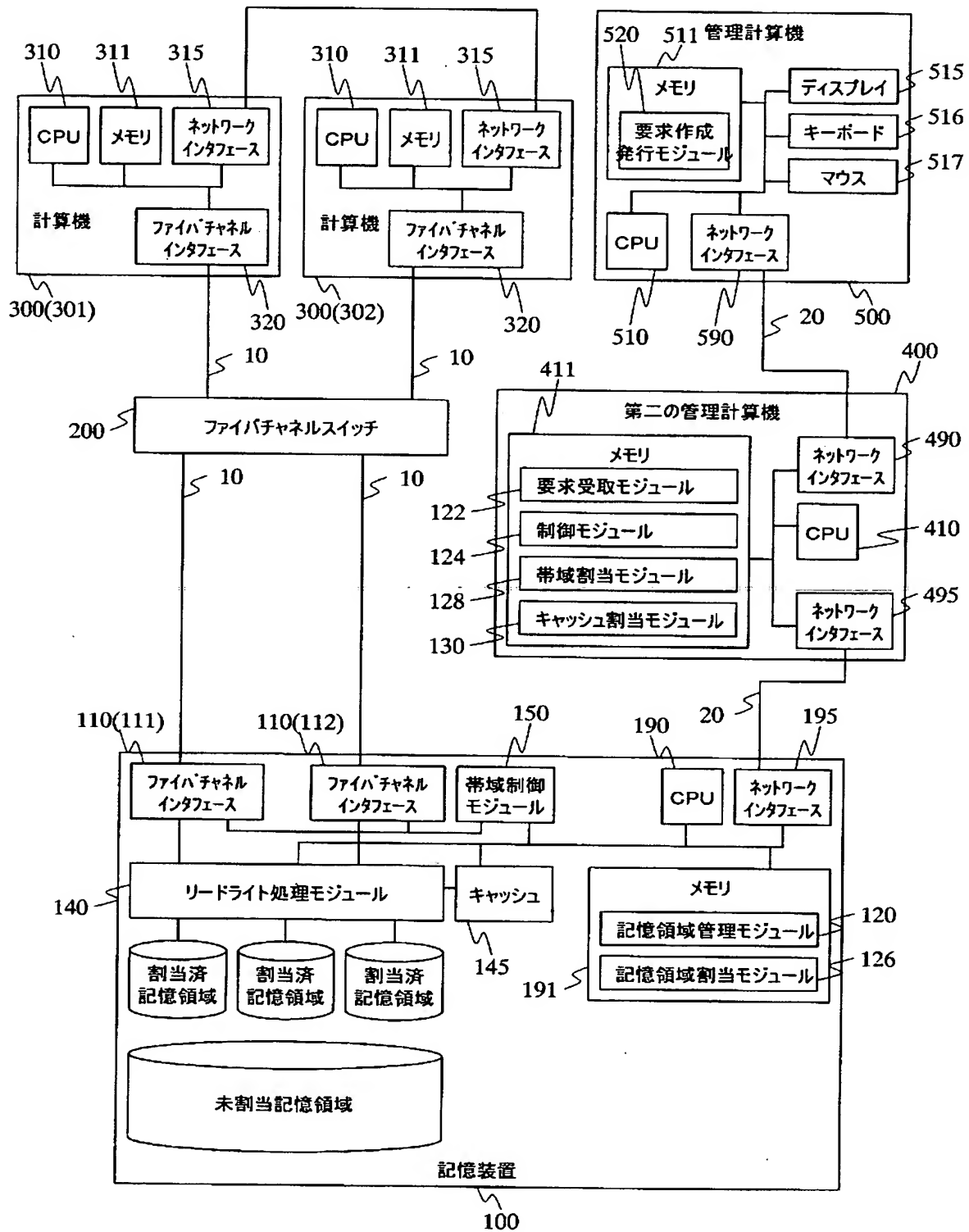
(B)

139

	記憶領域 番号	キャッシュ容量
割当済 キャッシュ	93	1GB
未割当 キャッシュ		3GB

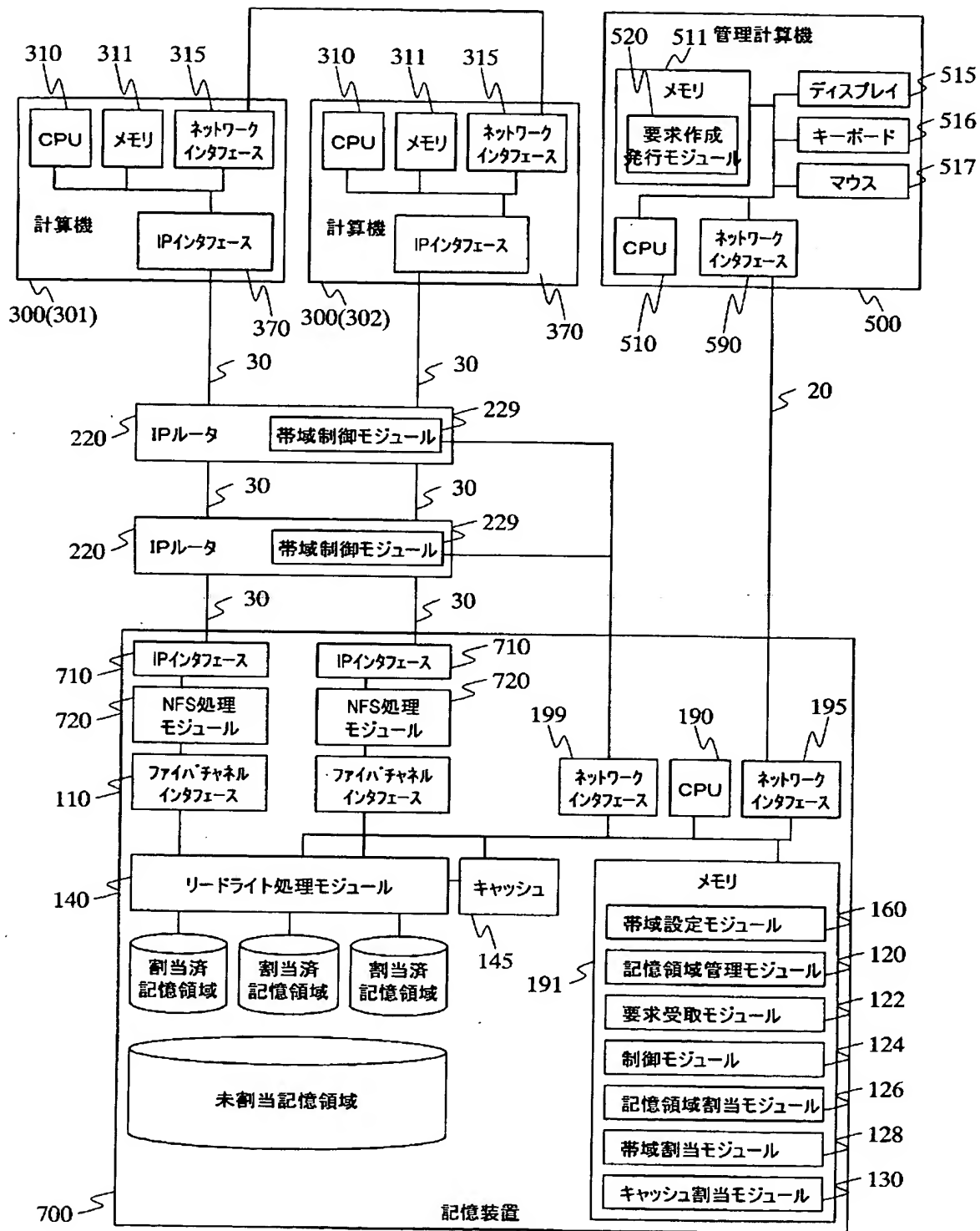
【図18】

図18



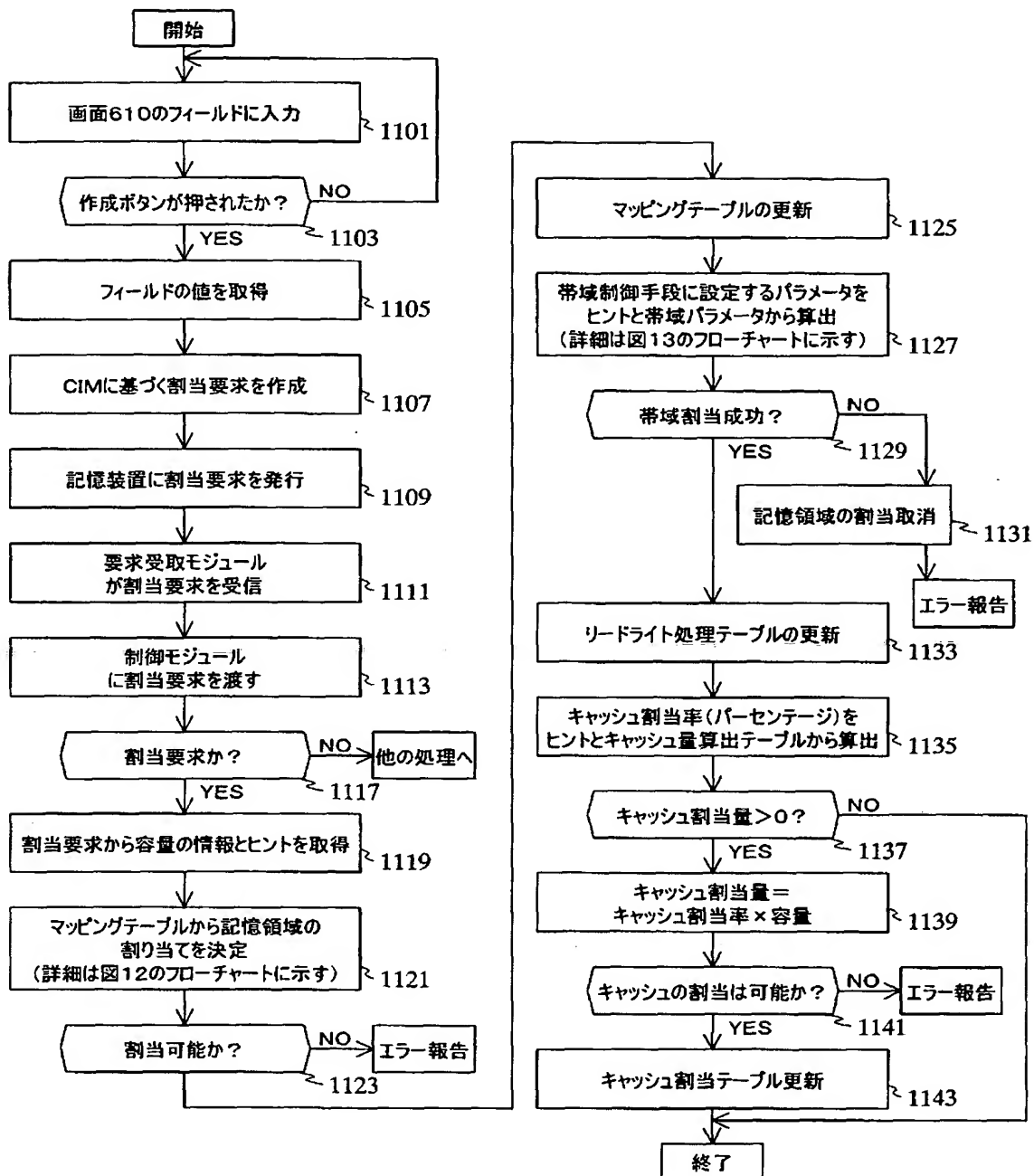
【図 19】

図19



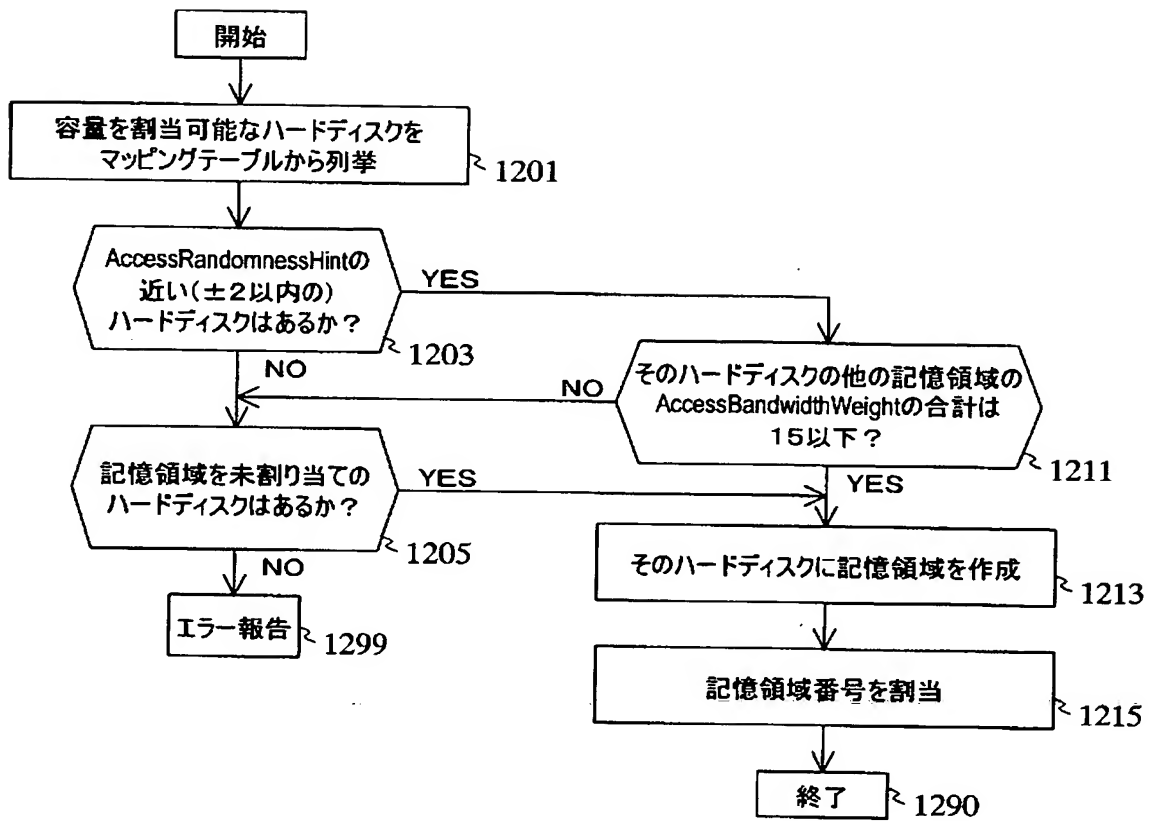
【図 20】

図20



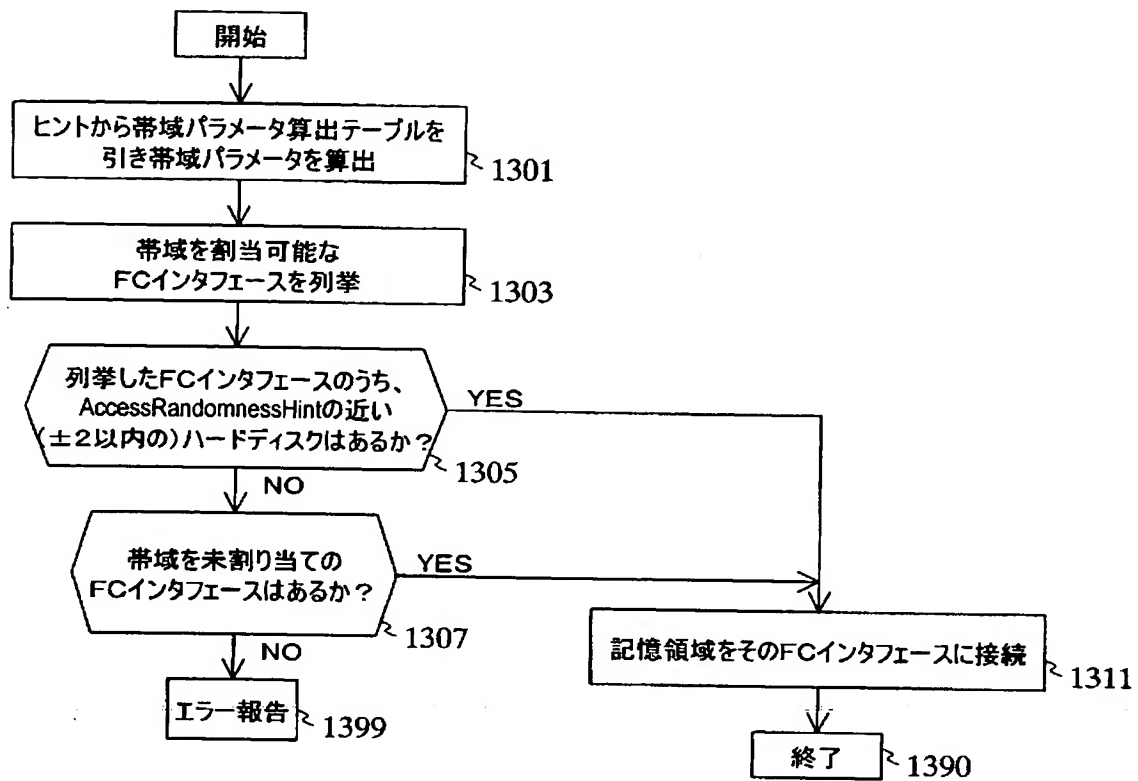
【図 21】

図21



【図 22】

図22



【図 24】

図24
(A)

171

FCインターフェース 番号	性能
111	1Gbps
112	1Gbps

(B)

172

FCインターフェース 番号	性能
116	2Gbps
117	2Gbps

【図 25】

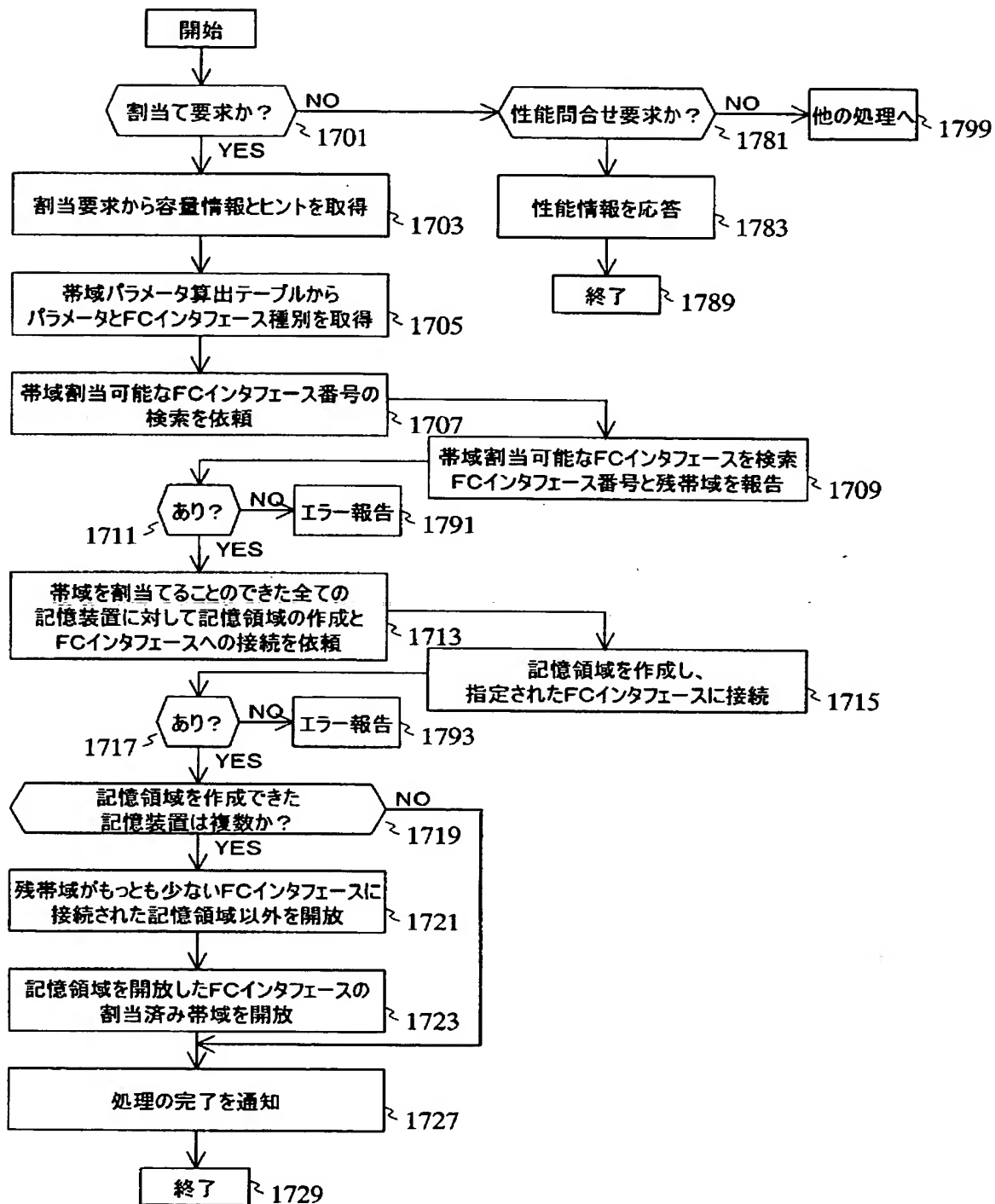
図25

129

Access Bandwidth Weight	帯域制御モジュールに 設定するパラメータ	
	1Gbps	2Gbps
0	10%	5%
1	20%	10%
2	30%	15%
3	40%	20%
4	50%	25%
5	60%	30%
6	80%	40%
7	*	50%
8	*	60%
9	*	70%
10	*	80%

【図 26】

図26



【書類名】 要約書

【要約】

【課題】 従来技術では、記憶領域を作成する際に指定したヒントに合致するように、帯域制御の設定パラメータを選択する必要があった。

、【解決手段】

本発明の一実施形態では、計算機によりアクセスされるデータを保持可能な記憶領域と、計算機と記憶領域との間でのデータのアクセスを制御するアクセス処理モジュールとを有する。管理計算機から、計算機に前記記憶領域を割当てて要求を受信する管理インタフェースと、記憶領域割当要求に従って、計算機に未割当の記憶領域を計算機によりアクセス可能なように割当てて記憶領域割当モジュールと、記憶領域割当要求に従って、計算機と割当てた記憶領域との間でのデータアクセスに影響する対象モジュールの性能を割当てて性能割当モジュールを有する。

【選択図】 図1

認定・付加情報

特許出願の番号	特願 2 0 0 3 - 1 8 9 1 5 7
受付番号	5 0 3 0 1 0 9 4 8 6 8
書類名	特許願
担当官	第七担当上席 0 0 9 6
作成日	平成 1 5 年 7 月 2 日

< 認定情報・付加情報 >

【提出日】

平成15年 7月 1日

特願 2003-189157

出願人履歴情報

識別番号

[000005108]

1. 変更年月日

1990年 8月31日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台4丁目6番地

氏 名

株式会社日立製作所